Data-Centric Energy Efficient Adaptive Sampling Techniques for Wireless Pollution Sensor Networks

by

Manik Gupta

A thesis submitted to the University of London in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

School of Electronic Engineering and Computer Science
Queen Mary University of London
United Kingdom

December, 2013

To my mother, who would have been proud

and

to my dear daughter Nitya, who makes me proud

Statement of originality

I, Manik Gupta, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

Signature:

Manik Gupta

Date: 31st March 2014

Details of collaboration:

CO pollution datasets were obtained in collaboration with UCL, London, UK and IIT, Hyderabad, India.

Abstract

Air pollution is one of the gravest problems being faced by modern world, and urban traffic emissions are the single major source of air pollution. This work is founded on collaboration with environmental scientists who need fine grained data to enable better understanding of pollutant distribution in urban street canyons. “Wireless sensor networks” can be used to deploy a significant number of sensors within a space as small as a single street canyon and capture simultaneous readings both in the time and space domain. *Sensor energy management* becomes the most critical constraints of such a solution, because of the energy hungry gas sensors. Hence, the main research objective addressed in this thesis is to propose novel temporal and spatial *adaptive sampling* techniques for wireless pollution sensor nodes that take into account the pollution data characteristics, and enable the sensor nodes to sample, only when, an important event happens to collect accurate statistics in as efficient a manner as possible. The major contributions of this thesis can be summarised as: 1) Better understanding of underlying pollution data characteristics (based on real datasets collected during pollution trials in Cyprus and India) using techniques from time series analysis and more advanced methods from multi-fractal analysis and nonlinear dynamical systems. 2)Proposal of novel adaptive temporal sampling algorithm called *Exponential Double Smoothing based Adaptive Sampling* (EDSAS) that exploits the presence of slowly decaying autocorrelations and local linear trends. The algorithm uses a time series prediction method based upon exponential double smoothing for irregularly sampled data. This algorithm has been compared against a random walk based stochastic scheduler called e-Sense and found to give better sampling performance. EDSAS has been extended to the spatial domain by incorporating distributed hierarchical agglomerative clustering mechanism. 3)Proposal of a novel spatial sampling algorithm called *Nearest Neighbour based Adaptive Spatial Sampling* (NNASS) that exploits the non-linear dynamics existing in pollution data to compute predictability measures to adapt the sampling intervals for the sensor nodes. NNASS has been compared against another spatial sampling algorithm called ASAP and found to give comparable or better sampling performance.

Acknowledgements

First and foremost, I would like to thank The Almighty for chalking out this path for me and guiding me on it. I would also then like to thank all the people, without whose contributions this work would not have been possible.

I express my sincere gratitude to Dr. Eliane Bodanese who provided me with the opportunity to pursue my PhD as part of the IU-ATC research project, and provided excellent direction and guidance, friendly challenge and thoughtful critique, encouragement and feedback throughout the duration.

I am grateful to Prof. Steve Hailes who has been instrumental in realising this research collaboration and also gave immensely useful research directions for the work. I am indebted to Dr. Venus Shum who has been a dear friend and an invaluable research mentor, providing useful suggestions and insights on this research work. My research collaboration with her has really been very effective and helped me tremendously improve this thesis.

A special thanks to Dr. Gareth Tyson for the proof-reading of this thesis and providing extremely useful inputs for improving the thesis.

This work would not have been possible without the immense love and support from my husband. I cannot be more thankful to him than anyone else for his constant patience and understanding during the challenging journey. Right from babysitting our daughter on the weekends, to participating with me in the late night debates on research nuances, debugging my algorithms, helping create my illustrations and proofreading this thesis – he has helped in every possible manner.

Finally, I am indebted to my dear family and lovely friends who have always stood beside me, helped me in so many little ways and provided with loads of encouragement in every situation.

Last but not the least, I would like to dedicate this work to my daughter - my little bundle of joy and inspiration - who while at times might have had to suffer the lack of attention from her mum being constantly on the laptop or being pre-occupied with the complexity of that clustering technique - but I hope someday she too will be inspired by this to do even greater work in whichever field she chooses to do.

Contents

List of Figures

List of Tables

List of Abbreviations

AAFT	Amplitude Adjusted Fourier Transform
ACF	Autocorrelation Function
ADC	Analog-To-Digital Converter
AP	Affinity Propagation
AR	Autoregressive
ARFIMA	Autoregressive Fractionally Integrated Moving Average
ARIMA	Autoregressive Integrated Moving Average
ARMA	Autoregressive Moving Average
ASAP	Adaptive Sampling Approach To Data Collection
ASSOC	Associate
BBQ	Barbie-Q A Tiny Model Query System
CH	Cluster Head
CO	Carbon Monoxide
CUSUM	Cumulative Sum
DFA	De-Trended Fluctuation Analysis
DFT	Discrete Fourier Transform
DSIC	Distributed Single Pass Incremental Clustering
DTW	Dynamic Time Warping
EDS	Exponential Double Smoothing
EDSAS	Exponential Double Smoothing Based Adaptive Sampling
EDSAS-S	Exponential Double Smoothing Based Adaptive Sampling - Spatial
EDSAS-T	Exponential Double Smoothing Based Adaptive Sampling - Temporal
EEDC	Energy Efficient Data Collection
EOFS	Environment Observation And Forecasting System

EPA	Environmental Protection Agency
ES	Exponential Smoothing
EWMA	Exponentially Weighted Moving Average
FFT	Fast Fourier Transform
GIS	Geographic Information System
HAC-DC	Hierarchical Agglomerative Clustering Based On Data Correlations
I^2C	Inter-Integrated Circuit
IU-ATC	India UK Advanced Technology Centre Of Excellence
KF	Kalman Filter
LEACH	Low Energy Adaptive Clustering Hierarchy
LRD	Long Range Dependence
MA	Moving Average
MESSAGE	Mobile Environmental Sensing System across Grid Environments
MFDFA	Multi-Fractal De-trended Fluctuation Analysis
MR	Miss Ratio
NAAQS	National Ambient Air Quality Standard
NN	Nearest Neighbour
NNASS	Nearest Neighbours Based Adaptive Spatial Sampling
PACF	Partial Autocorrelation Function
PAN	Peroxyacetylnitrate
PAQ	Probabilistic Adaptable Query System
PiP	Platform-in-Package
PLAMLiS	Piecewise Linear Approximation with Minimum Number Of Line Segments
ppm	Parts-Per-Million
PSE	Pervasive Sensing Environments

RCQ	Recursive Converging Quartiles
REP	Representative
RMS	Root Mean Square
RS	Rigenis Street
SAF	Similarity-Based Adaptive Framework
SES	Single Exponential Smoothing
SF	Sampling Fraction
SN	Sensor Nodes
SP	Sampling Performance
TE	Transfer Entropy
USAC	Utility Based Sensing and Communication
VN	Virtual Node
VOC	Volatile Organic Compounds
WAPMS	Wireless Sensor Network Air Pollution Monitoring System
WHO	World Health Organization
WSN	Wireless Sensor Networks

Chapter 1
Introduction

1.1 Research motivation

Sensor enabled smart devices, services and applications form the building blocks of the new era of ubiquitous computing in which today's human beings live. The sensor nodes in turn are the key enablers for providing such services and applications as they bridge the gap between the physical and the virtual world by collecting sensory data from the environment. Consequently, the field of Wireless Sensor Networks (WSNs) or the study of the organisation of these sensor nodes into distributed, smart self-organizing networks for the purposes of accurate data collection, smart data processing and efficient wireless communication, is an area of both large scale industrial application and active academic research and innovation.

WSNs are no longer just static network of sensor nodes, but they have evolved into highly dynamic networks of a large number of heterogeneous and mobile sensors capable of continuously capturing and processing a wide range of types of information that is precisely localized in the space and/or time, according to the user's needs or demands. These networks can even locally process – in a distributed fashion - the acquired information both at the unit and cluster level, and can also intelligently interact with the external world both by sending the outcome or aggregated features to the sinks and base-stations as well as receiving and reacting to the external inputs and commands.

In the industry, WSNs have already found wide application in many areas of crucial economic and social importance – agricultural research, healthcare delivery and monitoring, industrial process control, smart home monitoring, intelligent transportation, and environmental monitoring, to name just a few. Within each of these broad areas, WSNs are being used for a number of specific applications, for example, within environmental monitoring, WSNs are used for wildlife monitoring,

earthquake monitoring, landslide detection, air quality monitoring, tracking of environmental pollutants, water quality monitoring, forest fire detection, or other natural and man-made disasters etc. Researchers and architects have already developed and are actively deploying concepts such as “Smart Cities”, “Smart Homes”, “Smart Roads”, and “Smart Kindergarten”. It is not hard to imagine a day when human beings will be consciously or unconsciously interacting with some form of a WSN every single moment of their day and life.

While advances in sensor technologies continues to spur an ever increasing number of sophisticated applications of WSNs, some of its fundamental unique characteristics like limited energy, limited computing and memory resources, dynamic network topology, dense networks and possible heterogeneity of nodes do still continue to pose significant challenges in the design and development of novel algorithms and applications, despite a lot of research having been carried out to address these issues in the past two decades.

One such obstacle is “Energy consumption” particularly in the application scenarios where a dense deployment and a long network lifetime is expected. The sensor nodes are generally powered by batteries which have limited capacity and can neither be replaced nor charged easily due to environmental constraints. Most of the energy management schemes proposed in the literature have focussed on minimising the communication overheads as they assume that the energy consumption for data acquisition and processing is significantly lower than that of communication. However, this is not always true especially with the rising complexity of the applications. The same has also been recently highlighted that for many practical application scenarios, sensors consume significant amount of power [5],[6]. Hence energy management techniques that aim at minimizing only the radio activity are not sufficient to address the energy saving issue and need to be complemented with energy saving technique at the sensor level. These techniques called *sensor energy management* operate to reduce the number of data samples rather than the number of transmitted messages and form the backbone of this work, with the focus being on the design of novel techniques for sensor energy management in the temporal and spatial domain.

The current work was carried out under the India-UK Advanced Technology Centre of Excellence (IU-ATC) in Next Generation Networks, Systems and Services

[7] and was a collaborative effort amongst several academic and industrial partners both in the UK and India. The main research theme was *Pervasive Sensor Environments* (PSE) and the main focus was on achieving a tangible output in the form of a real test bed drawing on the existing hardware and software expertise across the research partner's current research programmes. The application area identified was environmental monitoring and was intentionally selected as being of significant social and economic interest in both the UK and India.

Distributed pollution level measurement was of significant interest to both the nations and infact, air pollution caused by traffic is an urban problem that has attracted increasing attention in the recent years. The atmospheric pollutants emitted from traffic sources are responsible for both acute and chronic effects on the human health. According to the World Health Organization (WHO) [8], an estimated 1.3 million deaths worldwide per year happen due to urban outdoor air pollution. Since, air pollution is a major source of health problems; it is an important issue in urban planning. At present, pollution monitoring is expensive and spatially coarse-grained, as it uses a few pollution monitoring stations at a few predetermined fixed locations. Specifically, it is of insufficient spatial granularity to allow for the development of predictive models that are capable of capturing pollution distribution on a street-by-street basis. Since the pollution levels can vary significantly on small geographic and temporal scales, fine-grained pollution measurement is necessary if both the health effects on individuals are to be estimated and the effects of proposed urban developments on local pollution levels are to be understood.

The focus of this research work, particularly, has been on the monitoring of pollution levels due to the car exhaust gases in urban street canyons; which are narrow streets flanked by tall buildings on both sides with little or no gaps in between. This leads to poor ventilation and as a result, the heat and pollutants are trapped at the street level. The pollutant concentration within a street canyon varies considerably in space and time and there are various factors like meteorological conditions, traffic emissions, building configuration etc. that impact the pollution dispersion. The environmental scientists need to get meaningful fine-grained pollution data to understand the complex pollution dispersion models [9],[10].

WSNs form a natural fit to the problem of fine grained pollution data collection, since, whilst individual sensor nodes may provide data of lower quality than the

pollution monitoring stations currently deployed, they are sufficiently cheap to allow their deployment in significantly larger numbers and hence provide high resolution temporal and spatial pollution data. The wireless nature of the sensor nodes means that the results can be viewed in real time, and the nodes can be deployed readily at positions determined by the scientists. Other advantage of the WSN system is the portability of the sensors. They can be carried by people to reflect true human exposure to traffic pollution, or be attached to vehicles or bicycles to get comprehensive and dynamic views of a larger area.

However, the trade-off in getting such fine-grained pollution data measurements via WSNs is that there will be a high energy demand on the individual resource constrained sensor nodes. Infact, the gas sensors used for measuring the pollution levels consume a lot of energy and therefore, sensor energy management is a topic very much relevant to the application at hand. Novel sensor energy management techniques (as focussed upon and proposed in this work) are needed such that the pollution sampling can reliably capture the most important events, while not losing the data fidelity. This will further enable more efficient data management services with only the most significant data being captured and analysed.

The algorithms proposed in this work are based on two fundamental hypotheses. One that some inherent characteristics of pollution data for example, trend and autocorrelation patterns, self-similarity, could be exploited to design data-centric adaptive sampling techniques and two that the pollution data from adjacent nodes could be highly redundant and exhibit both temporal and spatial correlations. Together, eliminating the acquisition of redundant data based on the above can therefore have a significant effect on the energy consumption of sensor nodes in a WSN. Infact the total reduction in the number of data samples can be translated into sensor energy savings using an appropriate sensor energy model. In this work, the sampled data reduction has been taken into consideration in order to evaluate the potential sensor energy savings.

So, it can be stated that the design and evaluation of novel sampling algorithms for pollution monitoring, that take into account the underlying pollution data characteristics, while maintaining the trade-off between the sampled data reduction and data fidelity, constitutes the main objective of this research work.

1.2 Main contributions of the thesis

Given the research objective of finding novel and innovative adaptive sampling schemes for pollution sensor networks, real pollution datasets were gathered during two pollution campaigns (carried out in Nicosia, Cyprus and Hyderabad, India) and studied to understand the underlying data characteristics. Pollution data are very dynamic in nature and appear significantly different from other datasets, and therefore require detailed analysis to understand the underlying behaviour. Various tools and concepts from time series analysis have been used to analyse the pollution datasets. Based on the insights gained from the data analysis, different data characteristics have been exploited for the design of temporal and spatial sampling algorithms proposed and evaluated in this work. So, the main contributions of this thesis are listed as follows:

1. *Application of time series analysis to fine-grained real pollution datasets*: Fine grained pollution data has been treated as a discrete time series and exploratory data analysis has been carried out using techniques from time series analysis [32] like trend and autocorrelation analysis. This was followed by an investigation for the presence of characteristics like long range dependence, self-similarity, multi-fractal scaling, non-linear dynamics [48] in these pollution datasets. Fine grained pollution datasets have been proven to possess interesting properties like multi-fractality and non-linear dynamics. The detailed data analysis enables a deeper understanding of the pollution data characteristics and guides the design of novel sampling techniques for pollution data.

2. *Proposition, design and evaluation of a novel adaptive temporal sampling technique*: A novel adaptive temporal data sampling algorithm called *Exponential Double Smoothing based Adaptive Sampling* (EDSAS) has been proposed. This algorithm exploits the presence of local linear trends and slowly decaying autocorrelations in the pollution datasets. It is based upon time series prediction method for irregularly sampled time series known as the *Wright's extension to Holt's method* [64] and incorporates a feedback mechanism based upon exponentially weighted moving average [66]. The performance metrics have been evaluated using the pollution datasets and the results compared against e-Sense [57], a random walk model based stochastic sampling scheduling algorithm.

EDSAS has shown better sampling performance in comparison to e-Sense for the different pollution datasets. The main advantages of EDSAS (design simplicity, low memory and communication overhead) have been highlighted as opposed to a model based adaptive sampling technique, which requires offline model construction and regular model updates/maintenance. EDSAS has also been applied to temperature and humidity datasets and shown to work well across range of different datasets.

3. *Application of EDSAS to the spatial domain*: EDSAS has been extended in the spatial domain to exploit the spatial data correlations to propose a novel spatial sampling algorithm called EDSAS-S. *Hierarchical agglomerative clustering* [77] based on data correlations have been incorporated as a pre-step to the spatial sampling to aid in energy conservation. Another recently proposed clustering approach called *affinity propagation* [78] has also been studied and applied to the spatial pollution data. The centralized affinity propagation serves as a benchmark for clustering performance for the distributed hierarchical agglomerative clustering. Detailed performance analysis has been presented for both the two spatial clustering approaches as well as the impact on the sampling algorithm has been studied. Based upon the detailed investigations, the hierarchical agglomerative clustering algorithm has proven to be a low-overhead clustering mechanism in comparison to the affinity propagation clustering. Furthermore, the spatial sampling technique has been evaluated across different algorithm parameters and found to provide substantial savings in terms of both the sensor energy consumption and the sampling message communication overhead.

4. *Proposition, design and evaluation of a novel adaptive spatial sampling technique*: A novel adaptive spatial data sampling algorithm called *Nearest Neighbours based Adaptive Spatial Sampling* (NNASS) has been proposed. It exploits the non-linear characteristics present in the pollution datasets. A non-linear measure, *predictability* [41],[42] has been proposed to be used in the design of the spatial sampling algorithm. The predictability measure is computed between different pairs of nodes to give an indication of how much one node's data can be predicted from another node's data. The nodes with higher predictability are sampled at lower sampling intervals in comparison to the nodes with lower predictability measure. NNASS has been investigated for sampling

performance in a clustered network architecture and the results have been presented in details for the different algorithm parameters. NNASS has also been compared against another spatial sampling algorithm called ASAP [85] and found to give better sampling performance.

1.3 Thesis structure

The structure of the thesis and the contents of the individual chapters are as follows:

1. Chapter 2 provides important concepts on WSNs and air pollution monitoring in order to set the context and enable an understanding of the work carried out in this thesis. An introduction to WSNs is given along with their applications for environmental monitoring. The major challenges in environmental monitoring are highlighted and the problem of sensor energy management [5],[6] is introduced in more detail. The typical mechanisms used for sensor energy management are explained. Also the description of facts and concepts on pollution dispersion in urban street canyons [9],[10] is given and how WSNs can help in achieving better pollution monitoring. A summary of other pollution monitoring research projects using WSN is also described in this chapter.
2. Chapter 3 provides the fundamental concepts on time series and techniques that are used for data analysis of pollution datasets. An introduction to the exploratory data analysis techniques like trend and autocorrelation analysis [32] is given. It is followed by a study of the more sophisticated techniques like time series forecasting [32](exponential double smoothing and autoregressive integrated moving average modelling), long range dependence, self-similarity and Hurst parameter analysis [33], multi-fractal analysis[34],[35] , time delay embedding [48] from the non-linear dynamical systems.
3. Chapter 4 presents details of the pollution trials carried out in Cyprus and India. The chapter presents the hardware and software [25], node deployment and placement details, as well as the sample pollution datasets gathered from the trials. This is followed by the application of time series concepts for data analysis of these real fine grained pollution datasets. The trend analysis, the autocorrelation analysis and the probability data distribution for pollution datasets are investigated. Further, the results of the application of different time series

forecasting methods, like the exponential smoothing and the autoregressive integrated moving average modelling, to these datasets is presented. Finally a Hurst parameter study using the multi-fractal de-trended fluctuation analysis is performed to confirm the presence of non-linear dynamics in pollution datasets. The selection of the time delay embedding parameters like the embedding dimension and the embedding delay is also investigated in this chapter.

4. Chapter 5 presents the proposition, the design and the evaluation of a novel temporal adaptive sampling algorithm called the Exponential Double Smoothing based Adaptive Sampling (EDSAS). A survey on the temporal adaptive sampling techniques in WSNs is provided along with an explanation of the reference algorithm, e-Sense [57],[58]. The basic principle behind the algorithm, Wright's extension to exponential double smoothing [64] for irregularly sampled time series is explained and the complete algorithmic design of EDSAS is presented. The performance metrics used to evaluate the sampling fraction and data measurement accuracy are explained. The algorithm hypothesis is tested using the pollution datasets, and the metrics are evaluated using various algorithm parameter values. Further, the EDSAS performance is compared against e-Sense and the drawbacks of a model based adaptive sampling technique are highlighted. EDSAS algorithm performance is also evaluated for temperature and humidity datasets.

5. Chapter 6 presents EDSAS-spatial (EDSAS-S) that is an extension of EDSAS to the spatial domain. The related work on various data collection and spatial sampling techniques that exploit spatial correlations in WSNs is presented. The spatial clustering mechanism is introduced to capture the spatial correlations existing between the closely located sensor nodes. Within each of the clusters, a representative node is chosen for sampling and the remaining nodes are put to sleep, whose sensing values are estimated using the data from the representative node. Two different spatial clustering algorithms: the hierarchical agglomerative clustering [77] based on data correlations and the affinity propagation [78] are investigated in detail and their clustering performance comparison is performed using the spatially interpolated datasets. The performance analysis of EDSAS-S in terms of sampled data reduction and data accuracy is carried out using the two different clustering algorithms and different algorithm parameters. In addition,

the messaging overhead accrued from the clustering and sampling is evaluated to understand the impact of the additional clustering process to the sampling algorithm.

6. Chapter 7 provides details on a novel spatial sampling algorithm called Nearest Neighbours based Adaptive Spatial Sampling (NNASS). Given the presence of self-similarity and multi-fractal characteristics in the pollution datasets, the self-similar nature can be used to compute a data measure called the predictability measure[41],[42], which indicates the degree of determinism in a time series. It can be further used to compute the level of interdependence between the data from different nodes. The predictability measures are computed amongst nodes in the clustered network architecture and used to assign adaptive sampling intervals to the sensor nodes. The performance evaluation of NNASS is performed using the spatially interpolated pollution data for the different algorithm parameters. NNASS is also compared against another spatial sampling algorithm called the Adaptive Sampling Approach to Data Collection (ASAP) [85] and the detailed results are used to evaluate the trade-off between the sampled data reduction and data accuracy are presented.
7. Chapter 8 provides a complete summary of the thesis and presents the future work that can be carried out both in the temporal and spatial adaptive sampling domains.

1.4 Author's publications

The research in this thesis has led to the following publications:

1. **Gupta, M.**, Bodanese, E., Shum, L. V., & Hailes, S. (2013, April). Exploiting nonlinear data similarities-a multi-scale nearest-neighbor approach for adaptive sampling in wireless pollution sensor networks. In *Proceedings of the 12th international conference on Information processing in sensor networks* (pp. 345-346). ACM.
2. **Gupta, M.**, Shum, L. V., Bodanese, E., & Hailes, S. (2011, October). Design and evaluation of an adaptive sampling strategy for a wireless air pollution sensor network. In *Local Computer Networks (LCN), 2011 IEEE 36th Conference on* (pp. 1003-1010). IEEE.

3. Shum, L. V., **Gupta, M.**, & Rajalakshmi, P. (2013). Data Analysis on the High-Frequency Pollution Data Collected in India. arXiv preprint arXiv:1301.7231.

Other research project related publications:

4. Shum, L. V., **Gupta, M.**, Bodanese, E., Karra, S., Glover, N., Malki-Epshtein, L., & Hailes, S. (2013, June). Bias adjustment of spatially-distributed wireless pollution sensors for environmental studies in India. In *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2013 10th Annual IEEE Communications Society Conference on* (pp. 318-326). IEEE.
5. **Gupta, M.**, and Eliane Bodanese. "Towards the Design of a Component-based Context-Aware Framework for Wireless Sensor Networks." *SENSORCOMM 2012, The Sixth International Conference on Sensor Technologies and Applications*. 2012.

Chapter 2
Wireless sensor networks and air pollution monitoring

2.1 Introduction

This chapter provides an overview and the necessary background for the two key domains that form the backbone of this research work – Wireless sensor networks (WSN) and air pollution monitoring.

By definition the current research work is quite interdisciplinary in nature and hence a good understanding of the key characteristics as well as the core issues relevant for both the domains is important to be able to put this work in the correct context. This also helps to understand why WSN should be used for air pollution monitoring and how sensor energy management techniques specific to the pollution domain can be formulated.

First, a brief introduction of WSN and their applications in environmental monitoring is provided. Various issues related to environmental monitoring using WSNs are discussed and the specific issue of sensor energy management is highlighted. An overview of the various approaches that can be used to address the sensor energy management problem is also presented. The second half of the chapter provides an overview of the complex phenomenon of air pollution dispersion and the multiple factors that impact the same. It is important to understand and take into account the factors that impact pollution dispersion, while designing an air pollution monitoring application.

The remaining of this chapter is organized as follows: Section 2.2 gives an introduction to WSNs and their applications in environmental monitoring are highlighted. The main limitations posed by WSNs are also listed. In the sub-section 2.2.2, the problem of sensor energy management is explored and the various techniques used to alleviate the same are explained. In Section 2.3, the characteristics

of pollution dispersion in urban street canyons and the reasons for using WSNs for pollution monitoring are explored. A survey of some other research projects that use WSNs for pollution monitoring is presented in Section 2.3.1. Finally, Section 2.4 draws the conclusions of the Chapter 2.

2.2 WSNs and challenges in environmental monitoring

2.2.1 Background

A WSN is usually composed of a large number of sensor nodes in the order of tens, hundreds or even thousands scattered in a sensor field with one or a few base stations/sinks, which connect the sensor networks to the users via the Internet or other networks. The nodes are deployed either inside the observed phenomenon or very close to it. The sensor nodes are equipped with sensing, data processing and communicating components to accomplish their tasks. Each of the sensor nodes is capable of collecting data and routing the data back to the sink by multi-hopping, as illustrated in Figure 2-1. The nodes can also organize themselves in clusters and cooperate to perform an assigned monitoring (and/or control) task without any human intervention. The sensor nodes sense the physical environmental information (for example, temperature, humidity, vibration, acceleration or whatever required), process locally the acquired data both at unit and cluster level, and send the outcome or aggregated features to the cluster head or base stations/sinks.

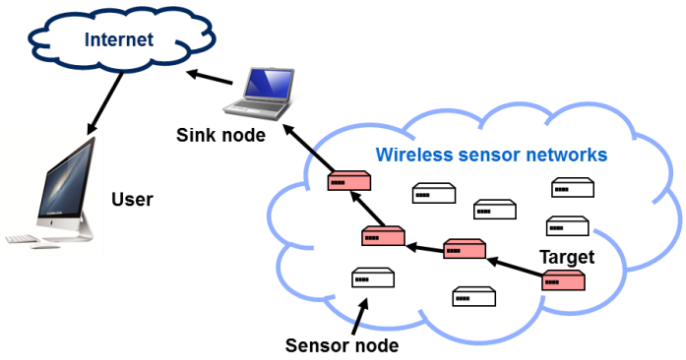


Figure 2-1 Typical wireless sensor network architecture

The work in this thesis focuses on a particularly type of sensor network application, environmental monitoring. Some of the main environmental applications of sensor networks include tracking the movements of birds, animals, and insects[1];

monitoring environmental conditions that affect crops and livestock; and environmental monitoring in marine, soil, glaciers [2] and atmospheric contexts; forest fire detection [3]; and flood detection [4]. While all these sophisticated WSN applications are emerging due to the advances in sensor technologies, WSNs still pose a lot of challenges due to their unique characteristics, for example, limited energy, constrained computing and memory resources, dynamic network topology, dense networks and possible heterogeneity of nodes. These constraints pose a significant challenge in the design and development of novel algorithms and applications for WSNs.

Energy consumption still remains one of the main obstacles particularly in application scenarios where a long network lifetime is expected. The sensor nodes are generally powered by batteries which have limited capacity and can neither be replaced nor charged due to environmental constraints. Though energy harvesting methods can be adopted to recharge the batteries, energy still remains a limited resource and needs to be used judiciously.

A typical sensor node consists of the following subsystems as shown in Figure 2-2 [5]:

1. A sensing subsystem includes one or more sensors (with associated analog-to-digital (A/D) converters) for data acquisition;
2. A processing subsystem includes a micro-controller and memory for local data processing;
3. A radio subsystem for wireless data communication; and
4. A power supply unit.

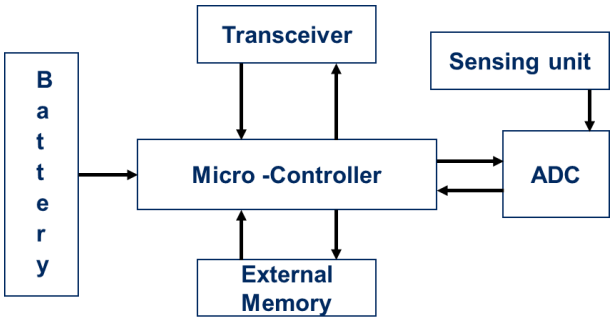


Figure 2-2 Typical sensor node architecture

So, the main energy consumption can be divided into three main domains – sensing, communication and computational processing. General observations with regards to different sensor node subsystems can be summarized as follows [5]:

1. The communication subsystem has much higher energy consumption than the computation subsystem. Therefore, communication should be reduced in order to achieve more energy savings.
2. The radio energy consumption is of the same order in the reception, transmission, and idle states, while the power consumption drops off in the sleep state. Therefore, the radio should be put to sleep (or turned off) whenever possible.
3. Depending on the specific application, the sensing subsystem might be another significant source of energy consumption, so its power consumption has to be reduced as well.

Most of the energy management schemes proposed in the literature assume that the sensing/data acquisition and processing have energy consumption significantly lower than communication. Recently, it has been highlighted that this assumption does not hold true for many practical application scenarios, where sensors consume significant amount of power [5]. A lot of factors like power hungry transducers, power hungry A/D converter, and longer acquisition times contribute to higher power consumption by the sensing subsystem. Hence energy management techniques which aim at minimizing the radio activity might not be sufficient to address the energy saving and need to be complemented with a energy saving technique at the sensor level. These techniques operate to reduce the number of data samples rather than the number of transmitted messages. As a part of this research work, the problem of *sensor energy management* has been addressed so as to reduce the actual number of sampled/sensed data points.

Fine grained monitoring of the pollution levels using energy hungry gas sensors can consume a lot of energy. Therefore, sensor energy management is a topic very much relevant to the application at hand and is explained in more details in the following section.

2.2.2 Sensor energy management

A general framework for sensor energy management is shown in Figure 2-3[5], [6]. Two main approaches can be considered to reduce the energy consumed by a sensor; they are duty cycling and adaptive sensing.

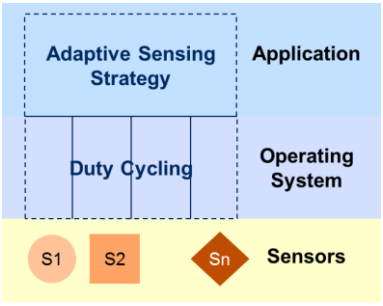


Figure 2-3 General framework for sensor energy management [5]

Duty cycling consists of waking the sensing system up only for the time needed to acquire a new set of samples and powering it off immediately afterwards. This strategy allows to optimally managing energy, provided that the dynamics of the monitored phenomenon are time-invariant and known in advance. The sampling rate is fixed and a synchronous equi-frequency sampling policy is adopted. The process dynamics are not taken into consideration and are assumed to be stationary which might not be the case in many applications. As a consequence, the sampling rate may be larger than necessary (over-sampling), leading to energy wastage or smaller than necessary (under-sampling), leading to the loss of important data characteristics. Duty-cycling schemes are oblivious to data that are sampled by sensor nodes. Therefore, it would be better to use application level data driven approaches to carry out energy efficient data acquisition.

An ***adaptive sensing/energy efficient data acquisition*** strategy which is able to dynamically adapt the sensor activity to the real dynamics of the process can be applied for sensor energy management. Adaptive sensing can be implemented using three different approaches [5],[6] as shown in Figure 2-4.

A brief description of each of the adaptive sensing approaches is provided below:

1. **Hierarchical sensing** requires units equipped with different sensors, each characterized by its own accuracy and power consumption, to measure the same physical quantity. At first, low-power sensors are considered to provide coarse

grained characterization of the sensing field and trigger an event. Then, accurate, but power hungry-sensors can be activated with measurements used to improve the coarser information. The idea behind hierarchical sensing techniques is to dynamically select which of the available sensors must be activated, by trading off accuracy for energy conservation.

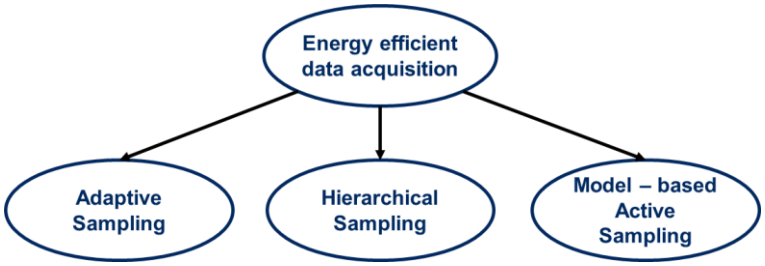


Figure 2-4 Classification of energy efficient data acquisition techniques [5]

2. **Adaptive sampling** techniques dynamically adapt the sampling rate by exploiting correlations among the sensed data. For instance, if the quantity of interest evolves slowly with time, so that subsequent samples do not differ very much, it is possible to take advantage of the temporal correlations. On the other hand, it is very likely that measurements taken by sensor nodes that are located spatially close each other do not differ significantly. Spatial correlations can be exploited to further reduce the sensing energy consumption.

3. **Model-based active sensing** builds a model of the sensed phenomenon on top of an initial set of sampled data. Once the model is available, the next data can be predicted by the model instead of sampling the quantity of interest, hence saving the energy consumed for data sensing. Whenever the requested accuracy is no longer satisfied, the model needs to be updated or re-estimated, to adhere to the new dynamics of the physical phenomenon under observation.

There are other data driven approaches [6] for energy conservation in sensor networks – ***data reduction*** including data compression and dual prediction techniques. Data compression can be applied to reduce the amount of information sent by source nodes. This scheme involves encoding information at nodes which generate data, and decoding it at the sink. Dual data prediction consists in building an abstraction of a sensed phenomenon, i.e. a model describing data evolution. The model can predict the values sensed by sensor nodes within certain error bounds, and reside both at the sensors and at the sink. If the needed accuracy is satisfied, queries

issued by users can be evaluated at the sink through the model without the need to get the exact data from nodes. On the other side, explicit communication between the sensor nodes and the sink is needed when the model is not accurate enough, i.e. the actual sample has to be retrieved and/or the model has to be updated.

Data-reduction schemes address the case of unneeded samples, i.e. the redundant information due to temporal and spatial correlations need not be communicated to the base station, while energy-efficient data acquisition/adaptive sensing schemes are mainly aimed at reducing the energy spent by the sensing subsystem, i.e. reduce the number of data points actually sampled by the sensors that is the main focus of the work of this thesis. Data driven adaptive sampling approaches that exploit both the temporal and spatial data correlations are the main focus of this research work. It is obvious that an efficient adaptive sampling strategy, by reducing the number of samples, also reduces the amount of data to be processed and transmitted to clusters and/or the base station or sink.

The next section explains the air pollution monitoring application in more details and examines why WSNs form the right technology for this application.

2.3 Air pollution monitoring using WSNs

This research work is founded on collaboration with environmental engineers interested in environment-related pollution dispersion modelling in urban street canyons [9],[10]. Hence, this section provides a background on the important characteristics of a street canyon that influence the air pollution dispersion levels. It provides a justification for the use of WSNs to carry out pollution studies at a micro environmental level.

Urban street canyons typify many dense urban environments and ideally refer to a relatively narrow street with buildings lined up continuously along both sides with few or no gaps between them. The dimensions of a street canyon are usually expressed by its aspect ratio, which is the height of the canyon divided by the width. The street canyons can be characterized as regular, avenue or deep canyon depending on the aspect ratio. Similarly, they can be classified as long, short or medium canyons depending upon the road distance between two major intersections. Urban streets might be also classified as symmetric canyons, if the buildings flanking the street have approximately the same height, or asymmetric, if there are significant

differences in building height. These differences in the street canyon geometry can affect the manner in which the pollutants get dispersed away. Because street canyons are particularly poorly ventilated and trap both heat and pollution at the street level; there is the possibility of changing urban planning practice to improve ventilation if the factors associated with it can be better understood.

One of the most characteristic features of the street canyon is the formation of a wind vortex [9],[10] due to the prevailing wind flow and as a result, the direction of the wind at street level is opposite to the flow above roof level as shown in Figure 2-5. These special wind flow conditions leads to a situation in which the pollutants emitted from traffic in the street are primarily transported towards the upwind side (leeward) while the downwind side (windward) is primarily exposed to background pollution and pollution that has re-circulated in the street. The strength of the wind vortices inside the canyon mainly depends on wind speed at roof-top level. However, the local wind flow is also affected by the mechanical turbulence induced by moving vehicles. The traffic induced turbulence is affected by the prevailing traffic volumes and average vehicle speeds.

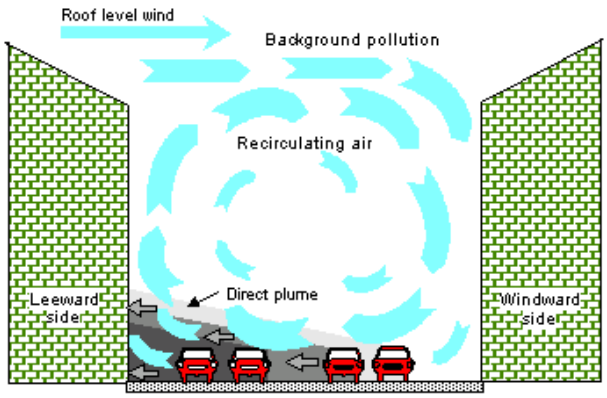


Figure 2-5 Illustration of wind flow and pollution dispersion in an urban street canyon [11]

Generally, the contribution to the pollution concentration at a point located at a distance x from the line source is given by [10] as follows:

$$dC_d = \sqrt{\frac{2}{\pi}} \frac{dQ}{u_b \sigma_z(x)} \tag{2-1}$$

where, Q is the emission in the street in $gm^{-1}s^{-1}$, u_b is the wind speed at the street level, and $\sigma_z(x)$ is the vertical dispersion parameter at a downwind distance x. It can

be seen from the equation (2-1) that the pollution concentration is proportional to the emission level, but is inversely proportional to wind speed and vertical wind pollution factor, which implies the stronger the wind speed, the lower the pollution level at the street level.

Theoretical work aimed at predicting formation of wind vortices is usually validated through simulation and modelling rather than from the collection of fine-grained real data due to the non-availability of such fine grained data to the environmental scientists. For example, water tunnels are constructed to validate the presence of wind vortexes modelled as a narrow street in Cyprus [12],[13]. WSNs with sensors to measure temperature, humidity, and wind speed in addition to pollution, provide an ideal addition to this set of tools for the environmental scientists. WSNs have the potential to provide multiple simultaneous data points at relatively fine temporal and spatial granularities and can help in the validation of models as well as to provide a primary source of information. Existing approaches to pollution monitoring typically use networks of few well-spaced high-quality monitoring stations. However, the dynamics of atmospheric pollution dispersion are such that even different sides of the same road, or locations at different heights above ground, can experience significantly different levels of pollution. The scarcity of good models means that it is difficult to estimate bounds on exposure across an area from single sample points. As a result, sensor systems that are distributed across an area are a natural way of obtaining direct measurements of the underlying spatio-temporal process and, consequently, in collecting information about the underlying physical processes, informing the construction of better models. WSNs make the acquisition of such data simpler and also enable the possibility of providing information (and alarms) in real time. The justifications for using the WSNs for pollution data collection are summarized as follows:

1. Measurement of average pollution levels: The pollution level (for example, carbon monoxide) depends strongly on the location and height of measurement even across a small space. Both the temporal as well as the spatial averages are as important to the environmental scientist to understand pollution dispersion at the micro environmental levels.
2. Understanding of the temporal dynamics of pollution: This can help to predict the traffic flows and congestion levels. The use of electrochemical sensors that give

near real time, fine-grained temporal information that can be relayed back to the users can help achieve this goal.

3. Snapshots of pollution level across the studied space: The snapshots of spatial pollution distribution changes with time and are required for the analysis of pollution hotspots, where pollution concentrates in an area due to the street structure and prevailing wind conditions.

Vehicular traffic generates a range of gaseous pollutants including carbon monoxide, carbon dioxide, nitrogen oxides, sulphur oxides, various volatile organic compounds (VOCs) such as aldehydes and peroxyacetylnitrate (PAN), and ozone. Of these, carbon monoxide has been identified by the US Environmental Protection Agency (EPA) in their National Ambient Air Quality Standard (NAAQS) [14] as a critical pollutant. It is neither photo reactive (unlike, for example, nitrogen dioxide) nor the result of photochemical reactions (unlike ozone or PAN), so its concentration is not dependent on the intensity of sunlight. Moreover, the natural background concentration for CO is of the order of 0.2 parts per million (ppm), with measured levels on the street of 5-20ppm, the difference being almost entirely due to engine exhaust trapped inside busy urban street canyons. The EPA NAQQS established the primary standard for CO as 9ppm for an eight hour average and 35ppm for a one hour average, not to be exceeded on more than one occasion per year. Consequently accurately measured CO concentrations correlate directly with the pollution input. This led CO to be selected as the primary pollutant of interest in this research work.

There are existing commercially available portable devices for CO monitoring, such as Learian ICOM devices used in [15] that provide time averages of CO readings where the data processing is performed under certain assumptions. These assumptions may not be valid in an outdoor street-level environment and interesting data characteristics may be lost in the process. Moreover, mostly regular or burst sampling techniques are used for collecting the pollution samples. However in this study, customized CO monitors [24],[25],[26] were designed and used in the pollution campaigns so as to provide low cost, well calibrated devices to gather highly accurate measurements. Fine grained CO data sampled at one second was collected during two different pollution trials in Cyprus and India across small cross-sections of urban streets. More details about the pollution trials can be found in Chapter 4.

The data obtained from per-second carbon monoxide (CO) measurements are highly dynamic, showing changes that can be attributed to individual vehicles or short timescale variation in traffic flows. Given this fine grained data, a post-hoc data analysis has been undertaken to understand the minimum (adaptive) sampling requirements needed to approximate pollution averages, and so to produce spatial snapshots of pollution dispersion and this forms the basis of the work carried out in this thesis. The main motivation for this analysis is to understand how sensor battery power can most effectively be conserved in longer term deployments, since the replacement of sensors in live environments is problematic.

2.3.1 Research projects on air pollution monitoring using WSNs

There are several pollution monitoring projects carried out using WSN technology, but they mainly focus on the networking, data collection and visualization aspects of the problem. One of the research projects, the Mobile Environmental Sensing System across Grid Environments (MESSAGE) [16]researched traffic management and route planning strategies using traffic and pollution related data.

Another example of urban air-quality monitoring is the CitySense test bed [18] that is developed by the researchers at Harvard University and BBN Technologies. The test bed consists of about a hundred wireless sensors mounted on buildings and streetlights across the city of Cambridge. Each node consists of a Linux based embedded PCs, with dual 802.11a/b/g radios and various sensors for monitoring weather conditions and air pollutants. CitySense is supposed to be a kind of experimental apparatus for urban-scale distributed systems and networking research efforts.

In [19], the authors have reported the work on the integration of a Geographic Information System (GIS) and a WSN system for an online ambient air monitoring system. The system is able to detect, measure and transmit information regarding the presence and quantities of internal combustion derived pollution and the geographical location in real time with the aim of creating pollution maps in urban environments.

The Environment Observation and Forecasting System (EOFS) [20] is an application for monitoring and providing a forecasting about environmental phenomena. An air pollution monitoring system is designed that involves a context model and a flexible data acquisition policy. The context model is used for understanding the status of air pollution on a remote place. It also supports the flexible sampling interval change for effective trade-off between the sampling rates and the battery lifetimes. This interval is changed depending on the pollution conditions derived from the context model.

The Wireless Sensor Network Air Pollution Monitoring System (WAPMS) [21] is a WSN air pollution system in Mauritius with a focus on the development of a data aggregation algorithm called Recursive Converging Quartiles (RCQ). The algorithm is used to merge data to eliminate duplicates, filter out invalid readings and summarise them into a simpler form that significantly reduce the amount of data to be transmitted to the sink and thus provide energy savings.

Wong et al. [22] undertook some interesting work in which they mounted sensors on vehicles, as opposed to using static sensors, with the aim of improving the spatial coverage of monitoring areas. Another example can be found in [23], where the urban pollution levels are monitored and mapped by means of an opportunistic mobile sensor network. The sensor equipment is installed on public transport vehicles, mainly buses, and the collected data is transmitted in different ways to a central repository and processing station.

So it can be seen that there have been several research projects that dealt with air pollution data collection, but none of them have focussed on data collection at fine resolution scales, while the main focus in the current work is on sampling the most useful and accurate information on pollution to be provided to the environmental scientist, while conserving the sensor battery power. The main objective in this research work is to come up with novel adaptive sampling strategies for pollution sensor networks that provide a balance between the sampled data reduction and data accuracy.

2.4 Chapter conclusions

This chapter has provided a background on WSNs and their applications in environmental monitoring. The problem of sensor energy management and the

various techniques used to tackle this problem have been explained. Adaptive sampling is one of the important techniques used for sensor energy management and it uses data correlations to adapt the sampling rate of sensor nodes. Adaptive sampling enables the sampling of only the most useful information from the environment and provides energy savings by avoiding the sampling of redundant data samples. Further along in this thesis, novel adaptive sampling techniques for pollution data are going to be investigated taking the peculiar pollution data characteristics into account.

Next, important preliminaries about air pollution dispersion have been discussed and how WSNs can enable better understanding of the environmental dynamics in urban street canyons. WSNs can provide fine resolution data at both the temporal and spatial scales for micro environmental studies. Finally, a survey of various research projects that have used WSN for air pollution monitoring has been given and the aim of the current research to collect accurate and fine gained pollution data was highlighted.

Chapter 3
Background on time series analysis concepts and techniques

3.1 Introduction

Before designing any new data-centric adaptive sampling techniques, it is important that the underlying pollution data characteristics be well understood. In this thesis, the pollution data is treated as a time series and specialised techniques from linear and non-linear time series analysis are employed to understand the data and isolate its particular characteristics. While the Chapter 4 presents the results of the data analysis, this chapter discusses the background on the various concepts and techniques from time series analysis.

The remaining of the Chapter 3 is structured as follows: Section 3.2 gives an introduction to the exploratory data analysis tools like the trend and the autocorrelation analysis. Section 3.3 gives an introduction to time series forecasting. Section 3.4 gives an introduction to the self-similarity and the long range dependence concepts. Section 3.6 explains the time delay embedding for phase space representation and the methods to select the embedding parameters. Section 3.7 lists the concluding remarks for this chapter.

3.2 Exploratory time series tools used for pollution data analysis

3.2.1 Definition of a time series

A time series is a sequence of observations that are arranged according to the time of their outcome [32]. Mostly these observations are collected at equally spaced, discrete time intervals. The characteristic property of a time series is the fact that the data are not generated independently, their dispersion varies in time, and are often governed by an internal structure (such as trend, autocorrelation or cyclic

components). This requires proper methods that are summarized under exploratory time series analysis. These can be used to analyse time series data in order to extract meaningful statistics and other characteristics of the data.

3.2.2 Stationarity

A time series is said to be stationary if there is no systematic change in the mean (no trend), if there is no systematic change in the variance and if strictly periodic variations are removed [32]. In other words, the properties of one section of the data are much like those of any other section. Most of the models are based upon stationary time series, and for this reason time-series analysis often requires the transformation of a non-stationary series into a stationary one. It is common to difference a given non stationery time series until it becomes stationary. *Differencing* [32] is a special type of filter where a new time series, say $\{y_2 \ldots . y_n\}$ is formed from the original observed series, say $\{x_1 \ldots . x_n\}$ by $y_t = x_t - x_{t-1}$ for $t = 2,3, \ldots . . n$. For non-seasonal data, first-order differencing is usually sufficient to attain apparent stationarity. Pollution datasets are observed to be non-stationary in nature and hence appropriate techniques need to be applied to study their behaviour.

3.2.3 Trend analysis

Trend in a time series is a slow, gradual change in some property of the series over the whole interval under investigation [32]. Trend is sometimes loosely defined as a long term change in the mean, but it can also refer to the change in other statistical properties.

There are no proven "automatic" techniques to identify trend components in the time series data. If the time series data contain considerable error, then the first step in the process of trend identification is smoothing using a filter. The most common technique is to use a moving average smoothing filter [32] that replaces each element of the series by either the simple or weighted average of n surrounding elements, where n is the width of the smoothing "window". Another method is to approximate the time series by fitting a linear function – in fact local linear trends can be fitted using a piecewise linear model [32] where the trend line is locally linear, but with change points where the slope and intercept change abruptly.

3.2.4 Probability distribution

The probability distribution of a time series describes the probability that an observation falls into a specified range of values. Probability distributions provide a visual aid for determining whether the time series is positively or negatively skewed.

3.2.5 Autocorrelation analysis

Autocorrelation refers to the correlation of a time series with its own past and future values. Autocorrelation is also sometimes called “lagged correlation” or “serial correlation”, which refers to the correlation between members of a series of numbers arranged in time [32]. Positive autocorrelation might be considered a specific form of “persistence”, a tendency for a system to remain in the same state from one observation to the next. Autocorrelation can be exploited for predictions: an auto-correlated time series is predictable, probabilistically, because future values depend on current and past values.

An important guide to the persistence in a time series is given by the series of quantities called the *sample autocorrelation coefficients*, which measure the correlation between observations at different times. Given n observations $x_1 \dots\dots\dots\dots x_n$ for a time series, the correlation between observations separated by k time steps or lag is given by[32]:

$$\rho_k = \frac{\sum_{t=1}^{n-k}(x_t - \bar{x})(x_{t+k} - \bar{x})}{\sum_{t=1}^{n}(x_t - \bar{x})^2} \tag{3-1}$$

The quantity ρ_k is called the autocorrelation coefficient at lag k and $\bar{x}$ denotes the overall mean. The set of autocorrelation coefficients arranged as a function of separation in time is the sample *autocorrelation function* (ACF). The plot of the autocorrelation function as a function of lag is also called the *correlogram*.

Another useful method to examine serial dependencies is to examine the *partial autocorrelation function* (PACF) - an extension of the autocorrelation, where the dependence on the intermediate elements (those within the lag) is removed. The PACF at lag k , denoted by π_k, may be interpreted as the correlation between x_t and x_{t+k} with the effect of the intermediate variables $x_{t+1} \dots\dots\dots x_{t+k-1}$ “filtered out”. In a sense, partial autocorrelation provides a "cleaner" picture of serial dependencies

for individual lags (not confounded by other serial dependencies). The calculation of PACF at various lag values $(0,1,2,3 \dots k)$ is depicted by means of the following equations:

$$\pi_0 = corr(x_0, x_0) = 1, t = 0, k = 0 \tag{3-2}$$

$$\pi_1 = corr(x_1, x_0) = \rho_1, t = 0, k = 1 \tag{3-3}$$

$$\pi_2 = corr\big(x_2 - \hat{E}(x_2|x_1), x_0 - \hat{E}(x_0|x_1)\big), t = 0, k = 2 \tag{3-4}$$

$$\pi_3 = corr\big(x_3 - \hat{E}(x_3|x_2, x_1), x_0 - \hat{E}(x_0|x_1, x_2)\big), t = 0, k = 3 \tag{3-5}$$

$$\pi_k = corr\big(x_{t+k} - \hat{E}(x_{t+k}|x_{t+k-1}, \dots, x_{t+1}), x_t - \hat{E}(x_t|x_{t+1}, \dots, x_{t+k-1})\big) \tag{3-6}$$

where, $\hat{E}(x_{t+k}|x_{t+k-1}, \dots, x_{t+1})$ denotes the linear regression of x_{t+k} on $x_{t+k-1}, \dots, x_{t+1}$.

Given n pairs of observations of two variables x and y, say $\{(x_1, y_1), (x_2, y_2) \dots \dots \dots (x_n, y_n)\}$, the *sample correlation coefficient* is given by[32]:

$$\rho_{x,y} = \frac{\sum(x_t - \bar{x})(y_t - \bar{y})}{\sqrt{\sum(x_t - \bar{x})^2 (y_t - \bar{y})^2}} \tag{3-7}$$

This quantity lies in the range $[-1,1]$ and measures the strength of the linear association between the two variables. The correlation is negative if high values of x tend to go with low values of y. If the two variables are independent, then the true correlation is zero.

The correlation coefficients are used in spatial sampling algorithm design in Chapter 6 to find out the relationship between the pollution data sensed from different sensor nodes. The correlation coefficients are one way of measuring relationship between different variables, but there are alternative data interdependence measures like the predictability measure that is described briefly in Section 3.5 and used in the design of a novel spatial sampling technique in Chapter 7.

3.3 Time series forecasting

A forecasting method is a (numerical) procedure for generating a forecast. When such methods are not based upon an underlying statistical model, they are termed heuristic. Forecasting using Exponential Smoothing is a heuristic method and it is described in details in Section 3.3.1. On the other hand, a statistical (forecasting) model is a statistical description of the data generating process from which a forecasting method may be derived. Forecasts are made by using a forecast function that is derived from the model. *Autoregressive integrated moving average* (ARIMA) models belong to this class of time series forecasting and are described in Section 3.3.2.

3.3.1 Forecasting using exponential smoothing

Exponential smoothing (ES) [32]refers to the general class of forecasting methods that rely on simple recursive equations to calculate the forecasts. The most basic form is called *single exponential smoothing* (SES) and it can be used for non-seasonal time series showing no systematic trend. Common to any ES technique, the measure is obtained by giving the highest weight to the latest observation and giving exponentially decreasing weights to the distant observations. A SES based forecasting, also known as *exponentially weighted moving average* (EWMA), has a basic recursive equation of the form of [32]:

$$L_{t+1} = \alpha y_t + (1-\alpha)L_t \qquad 0 < \alpha < 1, t > 0 \tag{3-8}$$

where, L_{t+1} is the next step estimate, L_t is the current EWMA and y_t is the latest reading. α is a smoothing constant that determines the weights given to the latest and historical data. Current data values are weighted more than historical values if α is greater than 0.5. In other words, the smoothed statistic L_{t+1} is a simple weighted average of the previous observation y_t and the previous smoothed statistic L_t. The one-step-ahead forecast from simple ES can be thought of as an estimate of the local mean level of the series, so that SES can be regarded as a way of updating the local level of the series. Equation (3-8) can also be written as[32]:

$$L_{t+1} = L_t + \alpha\varepsilon_t \tag{3-9}$$

where, ε_t is the forecast error for period t. In other words, the new forecast is the old one plus an adjustment for the error that occurred in the last forecast.

EWMA [32] can also be used as the basis of a change detection mechanism. Change detection is needed to identify any interesting changes happening in the time series. For example, in the case of a pollution monitoring application, pollution level changes going above a certain relative threshold (say 0.1 ppm) might need to be detected. EWMA based change detection can be used for this purpose. A long term moving average, L_{long} and a short term moving average, L_{short} can be calculated using SES based forecasting formula with two smoothing parameters, α_{long} and α_{short} respectively. The ratio $\eta = L_{short} / L_{long}$, provides an indication of a sudden change when η exceeds a threshold.

SES does not work well with datasets that exhibit trends. However, ES may readily be generalized to deal with time series containing trends. The version for handling a trend with non-seasonal data is usually called *exponential double smoothing* (EDS). EDS, as shown in equations (3-10) and (3-11), has a second smoothing parameter β for dealing with linear trends, which must be chosen in conjunction with α [32]:

$$L_t = \alpha y_t + (1-\alpha)(L_{t-1} + M_{t-1}) \qquad 0 < \alpha < 1 \tag{3-10}$$

$$M_t = \beta(L_t - L_{t-1}) + (1-\beta)M_{t-1} \qquad 0 < \beta < 1 \tag{3-11}$$

L_t represents the smoothed value at time t and M_t is the best estimate of the trend (i.e. the expected increase or decrease in the current level per unit time period) at time t. Equation (3-10) uses the current reading y_t and adjusts the smoothed estimate L_t using the previous period trend M_{t-1} and estimate L_{t-1} . Equation (3-11) then updates the trend M_t as the difference between the last two estimates. This double parametric form of EDS with two smoothing parameters α and β is also known as *Holt's method* [32]. The *k*-step forecast $\hat{y}_{t+k}$ is given by [32]:

$$\hat{y}_{t+k} = L_t + kM_t, \quad k \geq 1 \tag{3-12}$$

A variant of EDS that works for irregular time series is used for temporal adaptive sampling in this research work. The above procedure can be generalized to cope with seasonality in the data and is called the *Holt-Winter's* smoothing [32].

3.3.2 ARIMA modelling

Autoregressive moving average (ARMA) models [32] are mathematical models of the persistence, or autocorrelation, in a time series. An ARIMA model is a generalization of an ARMA model. ARIMA models are flexible and applied to a wide spectrum of time series analysis. These models are fitted to time series data to better understand the data or to predict future points in the series (forecasting). They are applied in some cases where data show evidence of non-stationarity, where an initial differencing step (corresponding to the "integrated" part of the model) can be applied to remove the non-stationarity. The model is generally referred to as an $ARIMA(p, d, q)$ model where parameters p, d, and q are non-negative integers that refer to the order of the autoregressive, integrated, and moving average parts of the model respectively.

3.3.2.1 Autoregressive (AR) Process

A stochastic process X_t is an autoregressive process of the order p, indicated by the notation $AR(p)$ [32]:

$$X_t = \alpha_1 X_{t-1} + \cdots + \alpha_p X_{t-p} + \in_t \tag{3-13}$$

where $\alpha_1, \alpha_2 \ldots \alpha_p$ are the parameters of the model and $\in_t$ is a white noise process.

Conceptually, an autoregressive process is one with a "memory", in the sense that each value is correlated with all preceding values. Following this interpretation, each value in an $AR(p)$-process is determined by p preceding values, where older values will have a fading effect. Low order processes, therefore, only have a "short memory". In $AR(1)$ process, also written as $ARIMA(1,0,0)$, the current value is a function of the preceding value, which is a function of the one preceding it, and so on. Thus each shock or disturbance to the system has a diminishing effect on all subsequent time periods.

3.3.2.2 Moving Average (MA) Process

A stochastic process X_t is a moving average process of the order q, indicated by the notation $MA(q)$ or $ARIMA(0,0,q)$ [32]:

$$X_t = \in_t + \beta_1 \in_{t-1} + \cdots + \beta_q \in_{t-q} \quad (3\text{-}14)$$

where $\beta_1, \beta_2 \ldots \beta_q$ are the parameters of the model and $\in_t$ is a white noise process.

The difference between an autoregressive process and a moving average process is subtle but important. Each value in a moving-average series is a weighted average of the most recent random disturbances, while each value in an auto-regression is a weighted average of the recent values of the series. Since these values in turn are weighted averages of the previous ones, the effect of a given disturbance in an autoregressive process dwindles as time passes. In a moving average process, a disturbance affects the system for a finite number of periods (the order of the moving average, q) and then abruptly ceases to affect it.

3.3.2.3 Differencing - Integration

A time series that reflects the cumulative effect of some process is called integrated. Such a time series has a trend and is non-stationary. The stationarity of a series is necessary for the estimation of AR and MA processes [32]. Therefore, time series that show a trend should be differenced, until stationarity is accomplished. In general, a first or second order differencing will be sufficient for series with a trend to become stationary.

3.3.3 Drawbacks of ARIMA modelling

Some major disadvantages of ARIMA forecasting are: first, some of the traditional model identification techniques for identifying the correct model from the class of possible models are difficult to understand and usually computationally expensive. Second, the underlying theoretical model and structural relationships are not distinct as the simple forecasts models such as simple exponential smoothing and Holt-Winters. Moreover, the ARIMA models, as all forecasting methods, are essentially “backward looking”. Such that, the long term forecast eventually goes to be a straight line and poor at predicting series with turning points.

3.4 Self-similarity and long range dependence

An exploratory time series analysis carried out on the pollution datasets leads to the intuition of the presence of self-similarity and long range dependence (LRD). These mathematical concepts are introduced below.

A discrete-time real-valued stochastic process $X = \{X_t, t = 0,1,2 \ldots\}$ is a strictly stationary process if all the distribution functions describing the process are invariant under a translation of time. X is said to be stationary in the wide sense, or weakly stationary, if the mean $\mu = E[X_t]$ is a constant, its variance $\sigma^2 = E[(X_t - \mu)^2] < \infty$ and its auto-covariance function depends only on the time difference k. The auto-covariance π_k is given by [33]:

$$\pi_k = Cov(X_t, X_{t+k}) = E[(X_t - \mu)(X_{t+k} - \mu)] \quad k = 0,1,2 \ldots \tag{3-15}$$

The auto-correlation function of X depends only on k and is given by [33]:

$$\rho_k = \frac{\pi_k}{\pi_{k=0}} = \frac{E[(X_t - \mu)(X_{t+k} - \mu)]}{\sigma^2} \quad k = 0,1,2 \ldots \tag{3-16}$$

Hence, for each k, ρ_k measures the correlation between elements of X separated by k units of time.

A wide-sense stationary stochastic process X is called a stationary process with long memory or long-range dependence if the autocorrelation function ρ_k satisfies the following relationship [33]:

$$\rho_k \sim k^{-\beta} \quad k \to \infty \tag{3-17}$$

where $0 < \beta < 1$. This implies that the autocorrelations decay to zero so slowly (in a hyperbolic manner) that their sum does not converge as k increases [33]:

$$\sum_{k=1}^{\infty} |\rho_k| = \infty \tag{3-18}$$

Intuitively, memory is built-in to the process because the dependence among observations that are widely separated in time is significant and the current observations retain some “memory” of the distant past. Another implication of this

non-summable autocorrelations is that, if n samples from the series are considered, then the variance does not decrease as a function of n but by a value of $n^{-\beta}$.

The simplest models with long-range dependence are self-similar processes. Self-similarity is one of the characteristics that result from a long-range dependency. Self-similar processes are particular attractive models because the long-range dependence can be characterized by a single self-similarity parameter, the *Hurst exponent, H*. A stochastic process X is self-similar if [33]:

$$X(at) = a^H X(t) \quad a > 0 \tag{3-19}$$

where the equality refers to equality in distributions, a is a scaling factor, and the parameter H is called the Hurst exponent.

Intuitively, self-similarity describes the phenomenon in which certain properties of the process are preserved irrespective of scaling in space or time. Self-similarity is the property associated with fractals - the object appears the same regardless of the scale at which it is viewed. A phenomenon that is self-similar looks the same or behaves the same when viewed at different degrees of "magnification" or different scales on a dimension. This dimension may be space (length, width) or time. There are several methods for estimating the self-similarity parameter H or the intensity of long-range dependence in a time series. One of these methods is the *de-trended fluctuation analysis* (DFA) and is described in the next sub-section.

3.4.1 De-trended fluctuation analysis

DFA is used to determine the scaling exponent of the signal that indicates the presence or absence of fractal properties (self-similarity) [34],[35],[36]. DFA is a scaling analysis method that provides a simple quantitative parameter to represent the autocorrelation properties of a signal. It is known for its robustness against non-stationarity.

DFA is typically employed to a time series with a random walk like structure. The time series is first converted to a random walk by subtracting the mean value and integrating the time series. The integrated time series is divided into non overlapping segments of equal duration, s. A polynomial trend is then fitted to the data within each time segment and local *root mean square* (RMS) is then computed for the

residual variation within each segment. The local RMS values, thus computed, for each of the time scale help to differentiate between the magnitudes of the local fluctuations. Then the q-order statistical moments are computed for the local RMS values over all the segments to obtain an overall q th-order RMS or a fluctuation function F_q [35]. Finally, the scaling behaviour of the fluctuation functions is analysed by log-log plots of F_q versus s for each value of q. In case of the presence of long range correlations, there exists a power law relation between F_q computed for multiple scales as follows [35],[36]:

$$F_q(s) = s^{H_q} \tag{3-20}$$

where, *q-order Hurst exponent, H_q* can be defined as the slope of the regression line, for each qth-order RMS. The existence of a power law distribution shows that data statistically exhibits self-similarity property at different observation scales. If H_q is independent of q, that is, H_q is a constant, then the time series is called *mono-fractal*. If H_q is dependent on q, the time series is called *multi-fractal*. When q has a large positive value, H_q implies large scale fluctuations. By contrast, if q is negative or has a very small positive value, H_q describes small scale fluctuations. The small scales are able to distinguish between the segments with high and low fluctuations (i.e. positive and negative q's, respectively) because the small scales are embedded within these segments. In contrast, the large scales cross several segments with both high and low fluctuations and will therefore average out their amplitudes.

The exponent $H_{q=2}$ is identical to the well-known Hurst exponent H [35],[36]. A larger Hurst exponent, H, is visually seen as slower evolving variations (i.e. more "structure") in the time series. A time series has a long-range dependent (i.e. correlated) structure when the Hurst exponent is in the interval of 0.5 to 1 and an anti-correlated structure when the Hurst exponent is in the interval 0 to 0.5. The q-order Hurst exponent H_q is only one of several types of scaling exponents used to parameterize the multi-fractal structure of a time series. Another q-order exponent, called the *mass exponent* t_q, [35],[36] can be computed from H_q as follows:

$$t_q = qH_q - 1 \tag{3-21}$$

The mass exponent t_q is often used to compute the *q-order singularity exponent* h_q and the *q-order dimension* D_q, [35],[36] as follows:

$$h_q = \frac{d}{dq}[t_q] \tag{3-22}$$

$$t_q = q(h_q) - D_q \tag{3-23}$$

A multi-fractal time series has mass exponent t_q with a curved q-dependency and consequently, a decreasing singularity exponent h_q. The plot of the q-order dimension D_q versus the q-order singularity exponent h_q is called the *multi-fractal spectrum*. The resulting multi-fractal spectrum is a large arc where the difference between the maximum and minimum h_q is called the multi-fractal spectrum width.

3.5 Predictability in time series

Time series predictability is a measure of how well future values of a time series can be forecasted, where a time series is a sequence of observations. Time series predictability indicates to what extent the past can be used to determine the future in a time series. A time series generated by a deterministic linear process has high predictability, and its future values can be forecasted very well from the past values. A time series generated by an uncorrelated process has low predictability, and its past values provide only a statistical characterization of the future values. Predictability can be evaluated between two series as well and can provide an alternative statistical quantitative interdependence measure.

Predictability improvement [41],[42] is a measure of the directional influences between the two time series. The predictability improvement successfully reflects the coupling strengths in both directions and allows detecting the asymmetry in the coupling information. It measures how much the future of one signal is clarified by the past of another signal. A nearest neighbour based approach is used for finding predictability measures [38],[39],[40] in this research work. It is explained in more details in Chapter 7.

3.6 Non-linear dynamics and time delay embedding

Recently, ideas from the science of nonlinear systems are being used to characterize and predict the dynamics of air pollution phenomena [43]-[47]. It has been revealed that, if the knowledge of the system in terms of its present and past states is available, then it is possible to make predictions of system's future behaviour. The approaches assume that it is possible to predict the future state of the system based on the single scalar time series assuming that, all the information regarding the external forcing factors is contained in that single time series.

The main idea in nonlinear time series modelling is that if the exact mathematical description of a dynamic system is unknown, the state space can be reconstructed from a single variable time series. The state space is defined as the multidimensional space whose axes consist of variables of a dynamic system. For example, for a three-variable model, the state space will be three dimensional and each of the three axes will be represented by a model variable. When the state space is reconstructed from a time series, rather than with actual model variables, it is customary to call this state space a *phase space* [48]. In general, a dynamic system can be described by a phase-space diagram whose trajectories describe the evolution of the dynamical system from some known initial states through time. In dissipative systems, (the ones in which the energy is not conserved), the trajectories eventually converge to some subspace regardless of the initial conditions. This subspace is called the *attractor* of the system [48]. In principle, the phase space contains the knowledge about the internal dynamics of the system and thus can be used as a predictive tool.

Time delay embedding [48] is the first step in the reconstruction of deterministic nonlinear dynamics from a time series. The dynamics of the system can be studied by studying the dynamics of the movement of the phase space points. The approach is useful for the cases, where the data on the explanatory variables influencing the time series of interest is not available.

Unfortunately, there is no generic way to select the best time delay embedding. Taken's embedding theorem [48] is very often invoked as the motivation for applying a time delay embedding to reconstruct multi-dimensional dynamics from a scalar variable. Let x_t be the scalar observed at integer times $t = 0,1 \ldots n$. The usual

incarnation of the time delay embedding is to obtain a vector of variables v_t such that [48]:

$$v_t = (x_t, x_{t-\tau}, x_{t-2\tau}, \ldots, x_{t-(m-1)\tau}) \quad (3\text{-}24)$$

and, by appealing to the theorem of Taken's, one claims that for a suitable τ and sufficiently large m and n the evolution of v_t is topologically equivalent to the underlying dynamical system. The reconstruction of a dynamical system from a time series requires the selection of two parameters, the embedding dimension m and the embedding delay τ. The *embedding delay* τ represents the separation of the nearby trajectories and the *embedding dimension* m represents the number of effective degrees of freedom available in the time series. The choice of both the parameters m and τ is crucial for the proper reconstruction of the attractor.

Many competing criteria to select these parameters exist, and all are heuristic. The reconstruction of a system dynamics usually begins with no knowledge of the number of state variables involved. There are therefore no time series available for each state variable. Generally, one starts with a measured time series consisting of a finite number of points sampled at equal time intervals.

Several methods are proposed in the literature to estimate the values of these parameters. To estimate τ, the methods based on an *autocorrelation function* and a *mutual information function* [48] are suggested. For selecting the appropriate value of τ, the autocorrelation function method is easy and involves relatively less computational time than the mutual information function. According to the shape of the autocorrelation function, the proper value of τ may be obtained. For example, a common choice for the time delay is the time, at which the autocorrelation function has its first minimum. One can also select that point as τ, where the autocorrelation function drops exponentially to 1/e or 1/10 or 0. Another approach based on the mutual information function is suggested in the literature [48] that takes into account nonlinear correlations. Accordingly, time delay τ can be obtained by using the mutual information function $I(\tau)$ defined as [44]:

$$I(\tau) = \sum_{i,j} p_{ij} \ln \frac{p_{ij}(\tau)}{p_i p_j} \quad (3\text{-}25)$$

where p_i is the probability to find a time series in the ith interval, and $p_{ij}(\tau)$ is the joint probability that an observation falls into the ith interval and the observation τ later falls into the jth interval.

For the estimation of m, the *false nearest neighbour* and the *correlation dimension* [44] based methods are available. In this work, the correlation dimension method has been used to estimate m. To complete the phase space reconstruction, the embedding dimension d_e, is estimated using correlation integral. For this purpose, in an d_e-dimensional space, the *correlation integral* $C(r)$ denoting the fraction of pairs with a distance smaller than a chosen radius r for increasing d_e is calculated as [43],[44]:

$$C(r) = \lim_{n\to\infty} \frac{1}{n^2 - n} \sum_{i \neq j}^{n} H(r - \|X_i - X_j\|) \tag{3-26}$$

where $H(.)$ is the *Heaviside function* that can be given as [43],[44]:

$$H(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases} \tag{3-27}$$

X_i and X_j are phase space vectors,$\|.\|$ is the Euclidean distance and r is the radius of the sphere centred on X_i or X_j . If the attractor for the time series data exists, then, for positive values of r, $C(r)$ is related to the radius r by the following relation [43]:

$$C(r)_{\substack{r\to 0 \\ n\to\infty}} \cong \alpha r^{\upsilon} \tag{3-28}$$

where α is a constant and υ is the *correlation exponent* or the slope of the $log\ C(r)$ versus $log\ r$ plot. If the correlation exponent is saturated with an increase in the embedding dimension m, then the process generating the time series is considered not random, rather deterministic. The saturation value of the correlation exponent is defined as the *correlation dimension* of the attractor, and the nearest integer above the saturation value provides the minimum number of the embedding dimensions of the phase-space required to model the dynamics of the attractor. For random processes, υ varies linearly with the increasing embedding dimension without arriving at a saturation value.

3.7 Chapter conclusions

This chapter has provided an in depth background of the major concepts of time series analysis that are required for understanding the data analysis carried out in the next Chapter 4. Details of various exploratory time series analysis concepts like the trend and the autocorrelation analysis that are essential to understand the data analysis of pollution datasets have been given. More advanced concepts like long range dependence, self-similarity, de-trended fluctuation analysis and time delay embedding from non-linear dynamical systems have also been explained in order to understand the non-linear characteristics in the pollution data.

Chapter 4
Experimental details and data analysis of pollution datasets

4.1 Introduction

As mentioned in the previous chapters, understanding the basic data characteristics of pollution data is fundamental to the design of adaptive sampling techniques. This chapter first describes the details of the pollution data collection experiments that were carried out in Cyprus and India. Then, it presents the results of analysis of the collected data using time series techniques, which supports the presence of data characteristics such as slowly decaying autocorrelations, self-similarity and non-linear dynamics.

It must be noted that this analysis of fine grained pollution data is one of the contributions of this thesis as generally, the environmental scientists do not have access to such fine grained data. The statistical analysis carried out in the atmospheric sciences literature is based on coarse grained measurements, for example, the authors in [37],[43] have used hourly average measurements. In contrast, in this research work, pollution data with high sampling rate of 1 sample/s was collected during the two pollution trials. The collected datasets are analysed and used for understanding the requirements for the sampling algorithm design and evaluation for pollution sensor networks.

The remaining of the Chapter 4 is structured as follows: the details on both pollution trials are given along with the information about the hardware and software used in the experiments in Section 4.2. Next, the results of the application of various time series analysis techniques to the real pollution datasets collected during the trials are presented in Section 4.3. The multi-fractal analysis is presented in Section 4.4 and the selection of the time delay embedding parameters is presented in Section 4.5. The

performed data analysis provides an insight into the pollution data characteristics and enables better algorithm design. Section 4.6 gives the conclusions of the chapter.

4.2 Pollution experiment details

Two different trials have been carried for pollution data collection – one in Cyprus and another in India. The sub-sections 4.2.1 and 4.2.2 give details about the hardware, software, node placement used in each of the trials and the pollution datasets collected thereof. The first pollution trials were carried out in Nicosia, Cyprus during April 2011 and the second pollution trials were carried out in Hyderabad, India during February 2012.

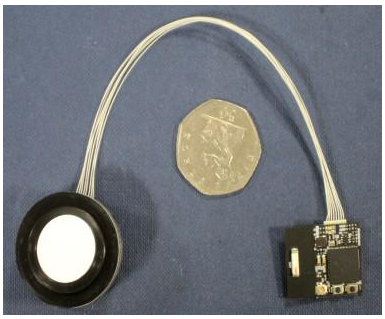
4.2.1 The details of the trials in Cyprus

A platform called Bracelet system [24], developed at UCL, has been used in the experiments. The Bracelet system can be tailored to specific applications according to the requirements. In-fact the flexibility of the Bracelet System enables to tailor the best solution based on cost and capability of the sensor nodes. The Bracelet system (a.k.a. Bracelet) customised for Carbon Monoxide (CO) monitoring used in the experiments in Cyprus is a low-cost modular system that has a low-power consuming Free-scale MC1322x Platform-in-Package (PiP)-based motherboard, which combines a 32 bit ARM7 processor with a built-in IEEE802.15.4 radio module, 96KB RAM and 128KB of flash memory that supports low power modes. The sensor module consists of a CO sensor, a bespoke signal conditioning circuit and a small microcontroller. The sensor module is connected to the motherboard via an Inter-Integrated Circuit (I2C) bus. Detailed description and references of the sensor modules and the Transducer Technology RCO100F electrochemical CO sensors used can be found in [25].

The sensor node was sampled at 1 sample/s and the readings were logged as 12-bit Analog-to-Digital Converter (ADC) values. The electrical output of the sensor (for example, current, voltage or resistance) passes into the signal conditioning interface, which converts the signal to an appropriately ranged voltage that is connected to an ADC value. The raw ADC values must be converted into part-per-million (ppm) for analysis by environmental engineers and this is done from individual calibration curves produced in a laboratory environment. More

information on the design of the devices and calibration results against temperature and concentration can be found in [26]. Figure 4-1(a) illustrates the Bracelet motherboard and sensor module, while Figure 4-1(b) shows the Bracelet CO monitor attached to a roadside lamp post during the experiments.

The Contiki operating system [27] was ported onto the Bracelet System. Rime [28], a simple networking protocol designed for WSNs within Contiki was implemented on the Bracelet and the collect mechanism in the Rime was used as the basis of multi-hop transmission of the data back to the base station in the network.

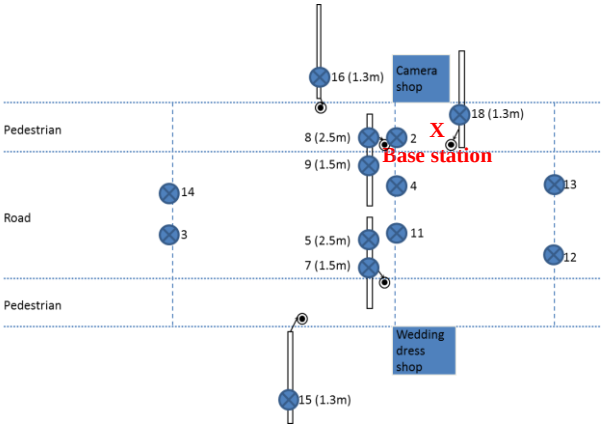


(a)

(b)

Figure 4-1 (a) Bracelet CO monitoring system (b) deployed CO monitor

(a)



(b)

Figure 4-2 (a) Area of deployment (b) node placement for Cyprus trial

The experiments were undertaken at Rigenis Street (RS), Nicosia, Cyprus as shown by the marked area in Figure 4-2(a). RS is a narrow one way residential/commercial street in Cyprus. It consists of 4 to 6 floor buildings, in which shops occupy the ground floors. The street is two-lane wide, with one parking lane and one way traffic. The traffic level was moderate during the 6-hour experiment,

with peak-hour traffic around mid-day and 3 to 4pm, with relatively frequent traffic jams. In this case, fourteen sensor nodes were placed on both sides of the street and at a range of heights (1.3m, 1.5m, and 2.5m) as shown in Figure 4-2(b). The nodes were able to form a multi-hop mesh network and deliver the data to the base station located close to the camera shop. They were attached to lampposts and cable ducts along the streets. In addition, CO monitors were also strung across the street at a height of two meters to measure the pollution level across the road.

One of the example pollution datasets collected from Node 7 is shown below in Figure 4-3. It shows data collected approximately from 10:00 am to 4:00 pm (five and half hours). It can be seen that the data is very spiky and there are few data points with very high ppm levels, though the mean pollution level is 1.78 ppm and the standard deviation is 1.57 ppm. The traffic-induced CO level exhibits themselves as sharp increases as the vehicles approached the sensors and sharp decreases as the vehicles moved away from the sensors. During traffic congestion, if there are high emission vehicles nearby the sensors, the local CO level accumulates and can be very high for a short period of time. Similar datasets were obtained from the other sensor nodes.

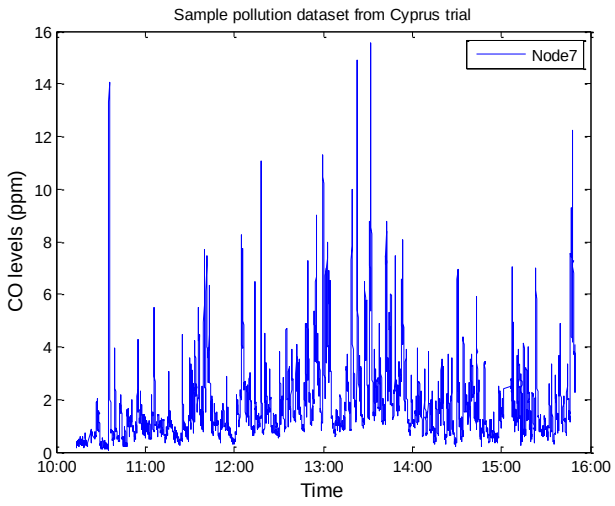


Figure 4-3 Sample dataset from Cyprus trial

Data from nodes 4, 7 and 16 are used to show the data analysis results in Section 4.3. These nodes are located at different locations and different heights in the street. Hence, they represent a range of data variations at the street level.

4.2.2 The details of the trials in India

The Bracelet wireless system used in the trials in Cyprus was further developed into the Orisen pollution monitors as shown in Figure 4-4 and they were deployed for t5he experiments carried out in India. Electrochemical carbon monoxide sensors RCO100F from KWJ Engineering Ltd. were used in the device. Each device also had temperature and humidity sensors (Sensirion SHT25) on board.

The CO monitors were equipped with Zigbee wireless capability, but they were not used in the experiments in India in order to extend the battery lifetime. In the experiment, the data was stored in SD cards, however, it is possible to achieve real time data collection and spatial collaboration with Orisen CO monitors. The CO data were collected at 1 sample/s.

Figure 4-4 Orisen CO monitors used in India experiments

These sets of data represent pollution levels at a wide motorway outside a shopping mall in front of a busy junction with 4 to 5 lanes of traffic in both directions. The pollution level measured was higher at this location than the trial in Cyprus. The experiment was conducted near Hyderabad Central Mall at the junction between Punjagutta Road and Nagarjuna Circle Road in Hyderabad, India as shown by the marked area in Figure 4-5(a). Fifteen monitors were scattered along a small cross-section of the road at different heights (1.5m and 2.5m) and locations in order to obtain fine grain spatial distribution as shown in Figure 4-5(b). Data from the fourteen devices were retrieved (one malfunctioned) with 28 hours data collected from 14:00 9/2/2012 to 18:00 10/2/2012.

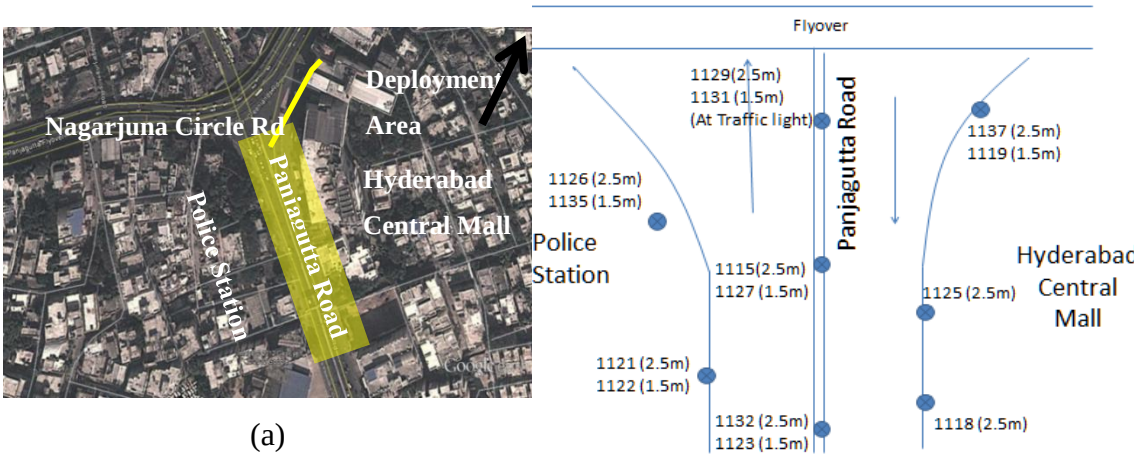


Figure 4-5 (a) Area of deployment (b) node placement for India trial

One of the example pollution datasets collected from Node 1119 is shown in Figure 4-6. The mean pollution level is 3.62 ppm and the standard deviation is 2.93 ppm. It can be seen that the pollution levels are significantly lower during the night time (10:00 pm to 6:00 am) as compared to the day time pollution levels that are quite dynamic and spiky in nature. Pollution levels with a maximum of 80 ppm have been measured in these experiments which are significantly higher in comparison to the pollution levels measured in the Cyprus experiments.

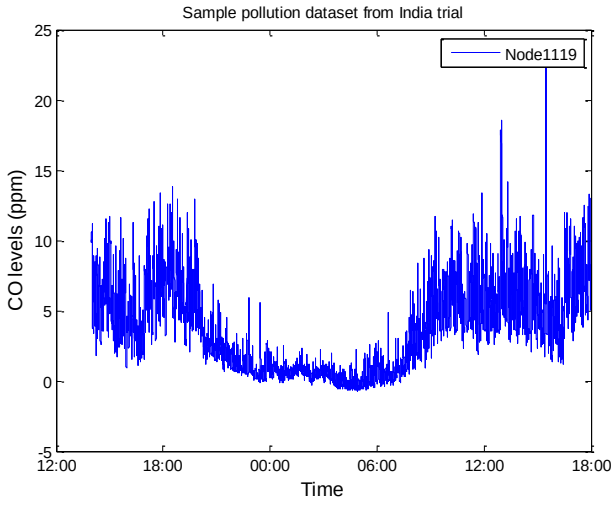


Figure 4-6 Sample dataset from India trial

Nodes 1119, 1122, and 1129 are chosen for data analysis in Section 4.3 since they represent a range of heights and locations across the busy street.

4.3 Exploratory data analysis for the pollution datasets

In this section results from the data analysis carried out on different pollution datasets from both trials are presented. The purpose of data analysis is to understand the nature of the data characteristics existing in the pollution data. The idea is to further exploit these underlying data characteristics for better sampling algorithm design and development. The results are shown for a few chosen sample nodes as mentioned in Section 4.2.1 and 4.2.2, but similar results are observed across the remaining nodes.

4.3.1 Data distribution of pollution data

Earlier research work, including [29] and [30] showed that pollutants are *log-normally distributed* at all the time scales and appropriate averaging time and frequency are proposed by the authors. Some of the results in these works have been used as guidelines for deriving pollution limits and standards. In [29], the authors studied the data distribution and averaging time between 5 minutes and 1 year and concluded that the averaging time had no effect on the sample distribution, which remained lognormal. A log-normal distribution is defined by:

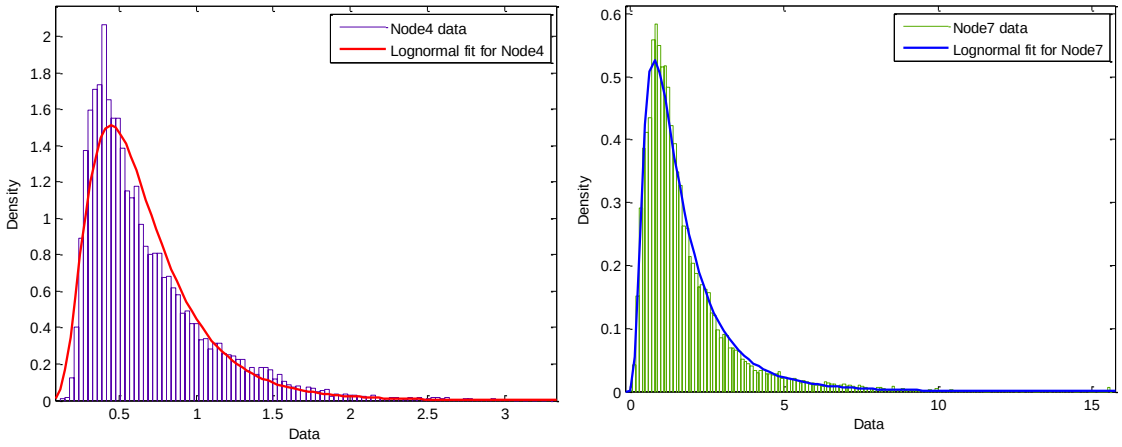
$$\ln(X) = N(\mu, \sigma^2) \tag{4-1}$$

where μ and σ^2 are the mean and variance of the logarithm of the time series. For a log-normal distribution, the mean and median of data are defined by:

$$\begin{aligned} Median(X) &= \exp(\mu) \\ Mean(X) &= \exp(\mu + \frac{\sigma^2}{2}) \end{aligned} \tag{4-2}$$

Hence, the parameter μ is associated with the median of the data, not the mean of the data. Log-normal behaviour is exhibited by 1 Hz pollution data obtained from both the trials as shown in Figure 4-7(a)-(b) and Figure 4-8(a)-(b). It should be noted that y-axis represents the probability density function and can be greater than 1. The properties of the probability density function are that it is nonnegative everywhere,

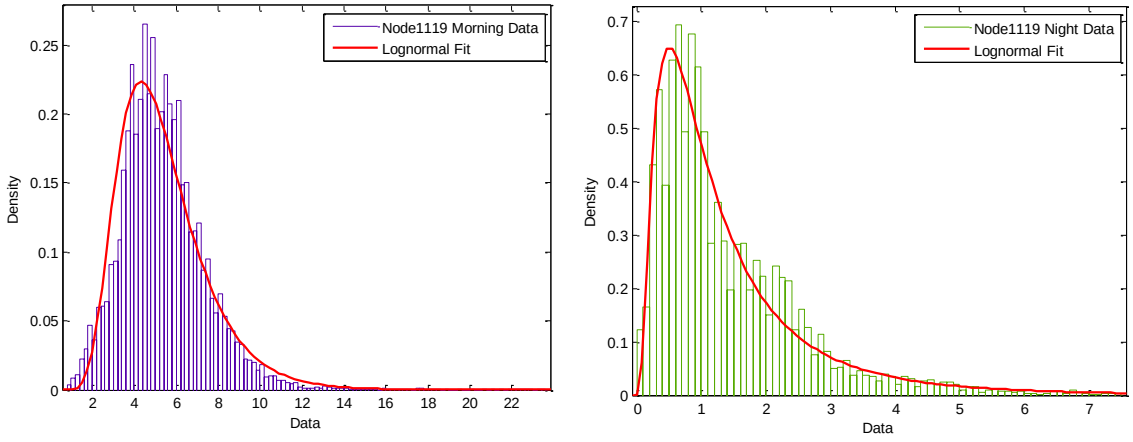
and the integral over the entire space is equal to one. With the high frequency Indian data, this lognormal distribution is observed both during the day and night time.



(a) (b)

Figure 4-7 Probability distribution of pollution data across few sample nodes from Cyprus trial

Log-normal distribution is one of the heavy tailed distributions where the right tail is longer and the mass of the distribution is concentrated towards the left. The log-normal distribution suggests that both the frequency of high-valued events and their duration is small compared to low-valued events. Based on the log-normal distribution, one of the requirements from the sampling strategy is to detect and capture the high values events or changes as much as possible.



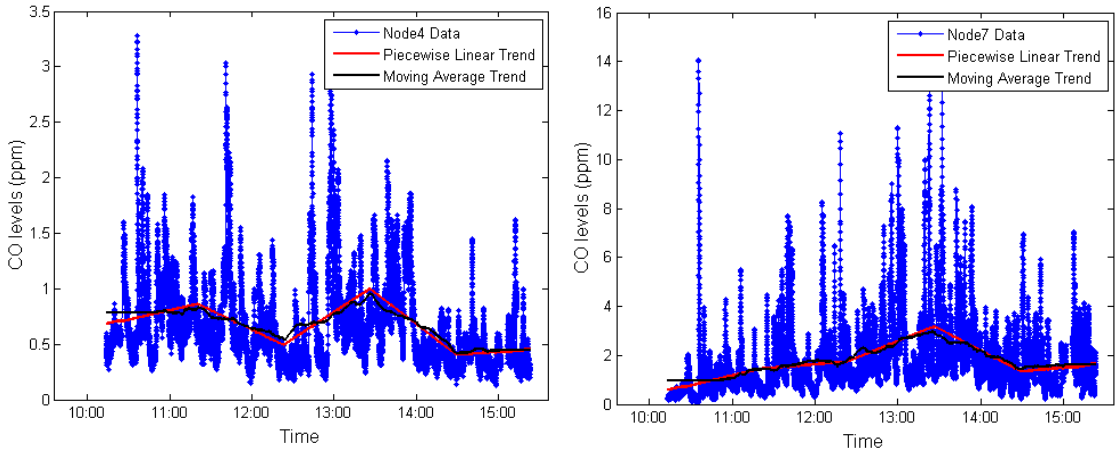
(a) (b)

Figure 4-8 Probability distribution for pollution data from India trial for (a) day (b) night time

4.3.2 Trend analysis

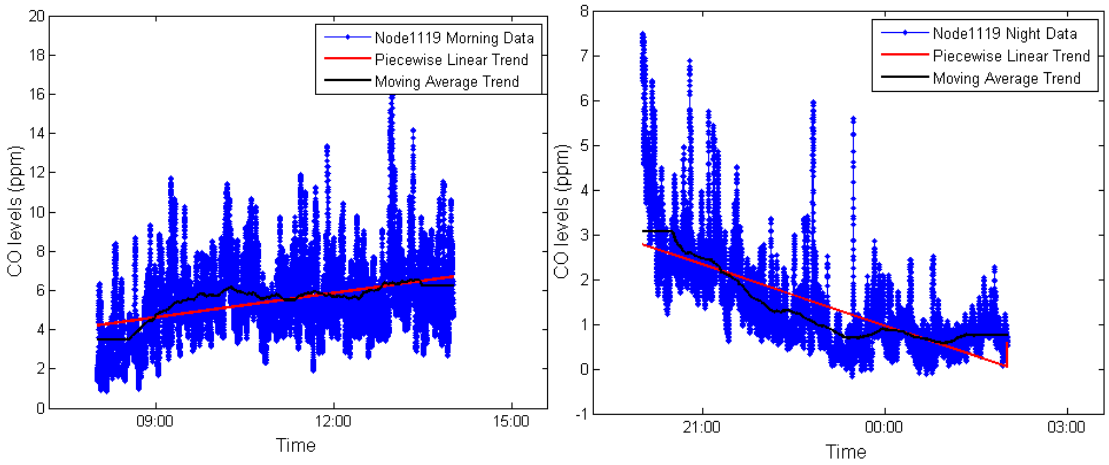
Next, it is important to understand and find out whether any significant trend or movement patterns exist in the pollution time series. The existence of trends in the time series can be proven using the moving average technique [32] and piecewise linear trend fitting [32] techniques as explained in Section 3.2.3. Here, both the techniques are applied to the pollution data from the Cyprus trial and the results for different nodes are shown in Figure 4-9(a)-(b). It can be seen from all these graphs that local linear trends do exist in the pollution datasets and can be exploited for algorithm development. The trend either increases or decreases linearly at small time scales (1 to 2 hours) for these datasets. The reason for varying trends during the short term is the varying levels of traffic and wind conditions.

Figure 4-10(a)-(b) shows the results for trend analysis applied to one of the node's data from India trials during the different times of the day. A distinct upward trend can be observed in the day time data whereas a downward trend is observed in the night time data. This can be attributed to the increased traffic levels and temperatures during the day time in comparison to the night time when both the traffic levels and temperature fall.



(a) (b)

Figure 4-9 Trend analysis for pollution data across few sample nodes from Cyprus trial



(a) (b)

Figure 4-10 Trend analysis for pollution data from India trial for (a) morning (b) night time

4.3.3 Autocorrelation analysis

Autocorrelations are measured by computing correlations between the data observations separated in time by k time steps or lags. It gives an idea about the persistence in the time series. Similarly, the partial autocorrelations are computed by removing the dependence on the intermediate data observations (those within the lag k). Autocorrelation and partial autocorrelation analysis are very important in order to understand the temporal correlations between consecutive data points and provides useful insights for the sampling algorithm design. More details on autocorrelations and partial autocorrelations can be found in Section 3.2.5.

The autocorrelation function (ACF) [32] and the partial autocorrelation function (PACF) [32] are computed for the different pollution datasets. It is observed that the ACF decays very slowly and does not become zero for a large number of time lags (i.e. the time separation between two different data points) – this suggests a high degree of persistence in the pollution datasets. The PACF analysis also indicates similarity of the data at very short time scales (less than 10s). Both these facts indicate that the data is very strongly correlated in the short time scales and this can be exploited to get additional sampled data reduction and thereby, achieve higher sensor energy savings. The results for both trials in Cyprus and India are shown in Figure 4-11 to Figure 4-14 for different times of the day and time scales (long and short duration).

Figure 4-11 shows the results from three different nodes 4, 7 and 16 for five hour data. The ACF is plotted against the time lags in the first column and the PACF is plotted against the time lags in the second column. It can be seen that even for time lags as high as 600s, the autocorrelations do not become zero. PACF shows a very strong lag-1 correlation of 0.99 and low order correlations (less than 0.2) for lags upto 10s.

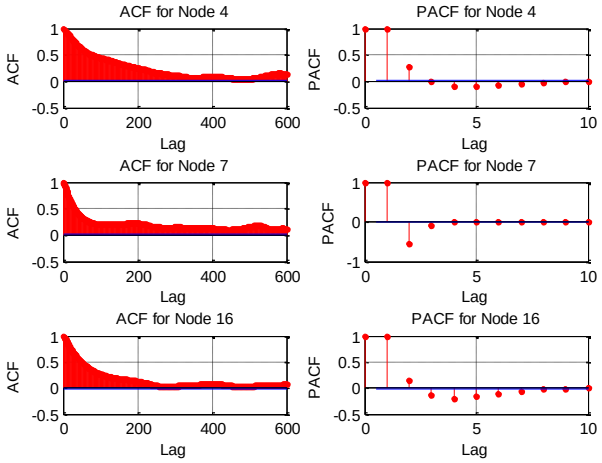


Figure 4-11 Autocorrelation and partial autocorrelation applied to few sample nodes from Cyprus trials for five hour data

The ACF and PACF can be computed at shorter time scales too and the results for 20 minutes of data for Cyprus data are shown in Figure 4-12. It can be observed that for node 7 autocorrelations become insignificant at 50s, while for the other nodes 4 and 16, data remains persistent till about 100s. This can be explained due to more dynamic data being sensed by node 7 than node 4 and node 16.

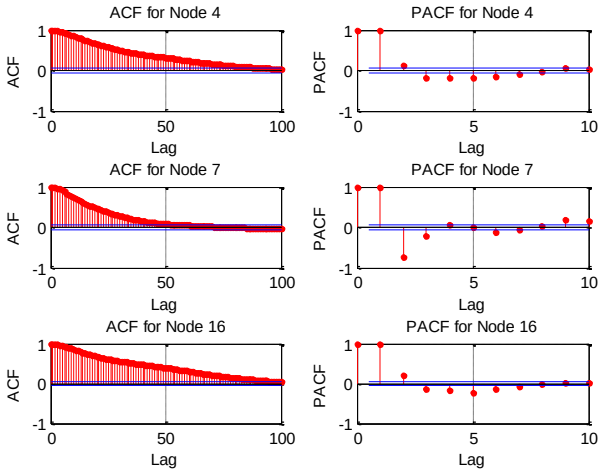


Figure 4-12 Autocorrelation and partial autocorrelation applied to few sample nodes from Cyprus trials for twenty minutes data

Similar results are observed for the Indian datasets as shown in Figure 4-13 and Figure 4-14. The ACF and PACF analysis is presented for six hour datasets. It can be seen that the autocorrelations do not decay till very high values of time lag. In addition, the night time data shows higher persistence as compared to the day time datasets. The autocorrelations do not become zero for lags as high as 600s for the day time data, but for the night time data, the lag values are much higher (more than 5000s). The PACF for the India data shows a very strong lag-1 correlation of 0.99 and lag-2 correlation of the order of 0.5. This can be attributed to similarity of data at short time scales (less than 10s).

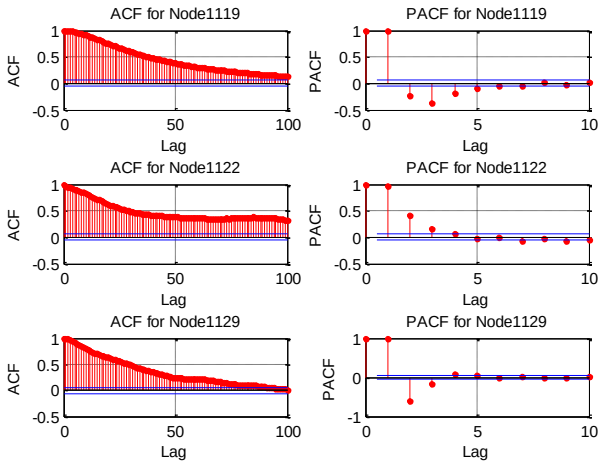


Figure 4-13 Autocorrelation and partial autocorrelation applied to few sample nodes from India trials for day time six hour data

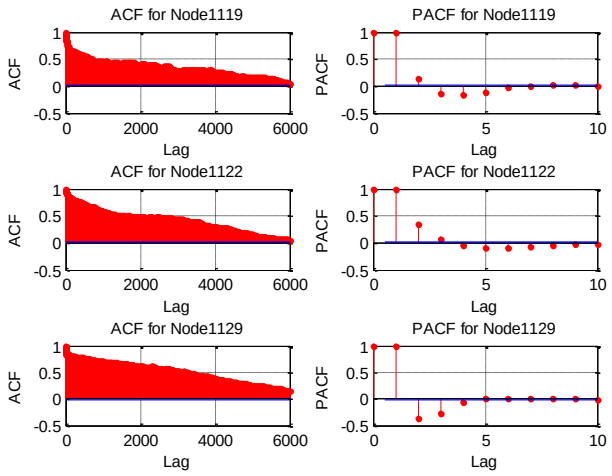


Figure 4-14 Autocorrelation and partial autocorrelation applied to few sample nodes from India trials for night time six hour data

4.3.4 ARIMA and EDS forecasting

ARIMA models [32] and EDS forecasting [32] are two different ways of time series forecasting and were explained in details in Section 3.3. Both time series forecasting techniques are applied to the pollution datasets using the R statistical tool [31] and the forecasting accuracy is compared using the *Box-Ljung* test [32].

The Box-Ljung test is a statistical diagnostic tool used to test the lack of fit of a time series model. The test is applied to the residuals of a time series after fitting the model to the data. The test examines the autocorrelations of the residuals and assesses the null hypothesis that a series of residuals exhibits no autocorrelation for a fixed number of time lags L, against the alternative that some autocorrelation coefficient $\rho_k, k = 1, \ldots, L$ is nonzero. The test statistic is given by [32]:

$$Q = n(n+2)\sum_{k=1}^{L}\left(\frac{{\rho_k}^2}{T-k}\right) \quad (4\text{-}3)$$

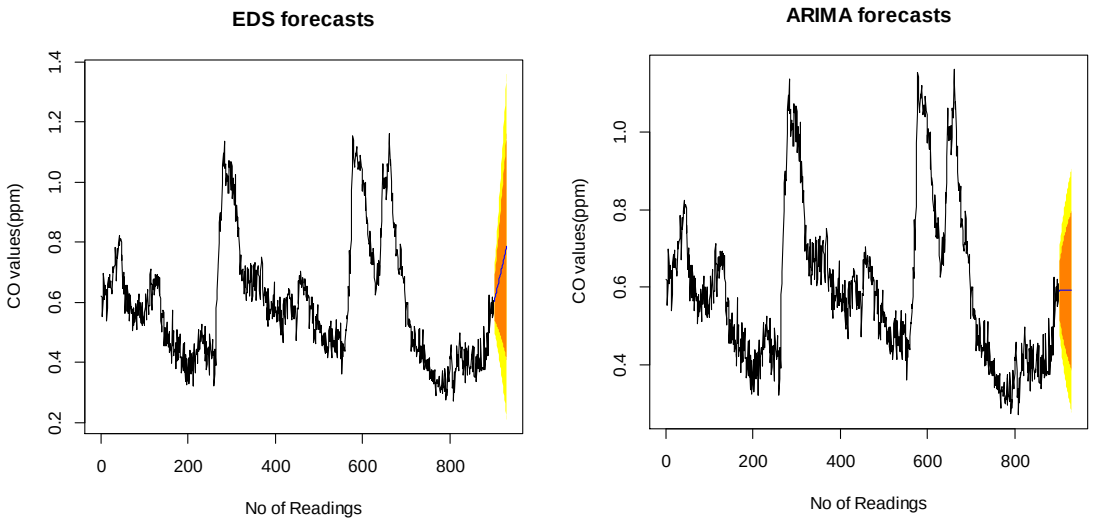
where, n is the sample size, L is the number of autocorrelation lags, and ρ_k is the sample autocorrelation at lag k. For a given significance level α, the critical region for rejection of the hypothesis of randomness is given by [32]:

$$Q > \chi^2_{1-\alpha,L} \quad (4\text{-}4)$$

where, $\chi^2_{1-\alpha,L}$ is the α-quantile of the chi-squared distribution with L degrees of freedom. The chi-squared distribution with L degrees of freedom is the distribution of a sum of the squares of L independent standard normal random variables. The significance level, α is set to 0.05. The test returns a *p-value* of the test statistic that indicates whether to reject the null hypothesis or not. If the p-value is less than 0.05, the null hypothesis can be rejected and this indicates that the autocorrelations are significant and the model does exhibit a lack of fit. On the contrary, if the p-value is greater than 0.05, it indicates that the autocorrelations are very small. It can be concluded that the model is appropriate and does not exhibit a significant lack of fit.

Figure 4-15(a) shows fifteen minutes of data from one of the Cyprus nodes (Node 4) and corresponding EDS forecasts for the next 30s.The forecasts are shown as a blue line, with the 80% confidence levels for the prediction values shown as an orange shaded area, and the 95% confidence levels for the prediction values shown as

a yellow shaded area. Similarly, Figure 4-15(b) shows the results of the application of ARIMA (2,1,1) model to the data. Once the forecasting had been performed,the *Box-Ljung* statistical test is applied and the autocorrelations for time lags 1-10 are tested in the in-sample forecast errors. The results are tabulated in Table 4-1.



(a) (b)

Figure 4-15 Confidence intervals for forecasts made using (a) EDS (b) ARIMA (2,1,1) model

Table 4-1 Box-Ljung results for ARIMA and EDS forecasting

	Box-Ljung test results for ARIMA
Lag	**p-value**
5	0.2341
10	0.05191
	Box-Ljung test results for EDS
Lag	**p-value**
5	0.7539
10	0.7802

It is found out that p-value for EDS forecast errors is almost 0.7 indicating that there are no autocorrelations in the in-sample forecast errors at time lags 1 to 10. On the other hand, the p-value for ARIMA forecast errors is almost 0.2 at lag set to 5, which indicates that no autocorrelations are present in in-sample forecast errors, but the p-value for lag set to 10 is almost 0.05, which indicates that there might be correlations in the forecast errors and the model does exhibit a lack of fit. The results

from the Box-Ljung statistical test prove that EDS forecasting performs better than ARIMA modelling for the pollution datasets and can be used for the algorithm design.

4.3.5 Summary of the exploratory pollution data analysis

Based on the trend and autocorrelation analysis, it can be seen that pollution datasets have locally linear trends and slowly decaying autocorrelations, i.e. the datasets exhibit high degree of persistence. ARIMA modelling is not suitable because of the complexity and regular model updates. Simpler recursive, real-time implementation of EDS forecasting can be exploited to give results better than ARIMA modelling. This data analysis forms the basis for the novel temporal sampling strategy proposed as a part of this research work. More details can be found in Chapter 5.

4.4 Multi-fractal de-trended fluctuation analysis (MF-DFA)

The autocorrelation analysis carried out in Section 4.3.3 suggests that the correlations are slowly decaying and hence the pollution datasets may exhibit long range dependence and hence self-similarity. Given the non-stationary nature of the pollution datasets, de-trended fluctuation analysis (DFA) [34],[35],[36] is used to study the behaviour and verify the presence of self-similar data characteristics. DFA is a method used to study the intensity of long range dependence and estimate the self-similarity parameter, the Hurst exponent in a time series. More details on the DFA technique can be found in Section 3.4.1. In this section, the results from the application of DFA on the pollution datasets are presented.

The local RMS values are computed for multiple time scales varying from 60s to 1200s using the DFA method and the magnitude of the local fluctuations are shown in Figure 4-16 for the Cyprus datasets. It can be observed that the fast changing fluctuations can be visualised at the smaller scales while the slow changing fluctuations are seen at the larger scales.

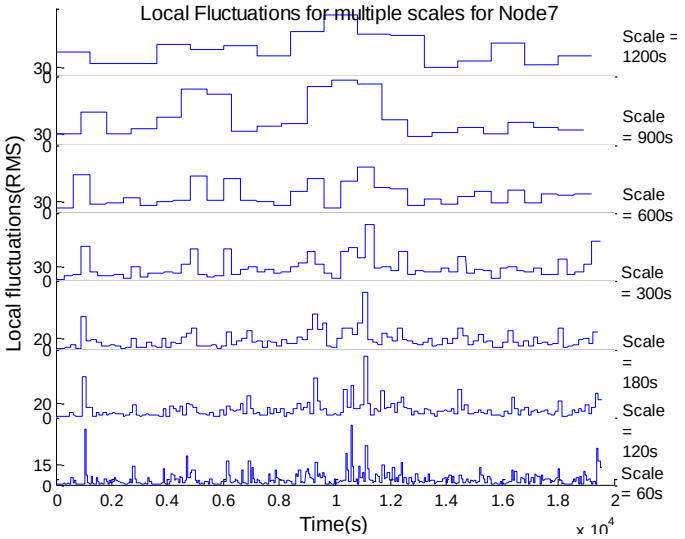


Figure 4-16 Local fluctuations in the pollution dataset for multiple scales for Cyprus data

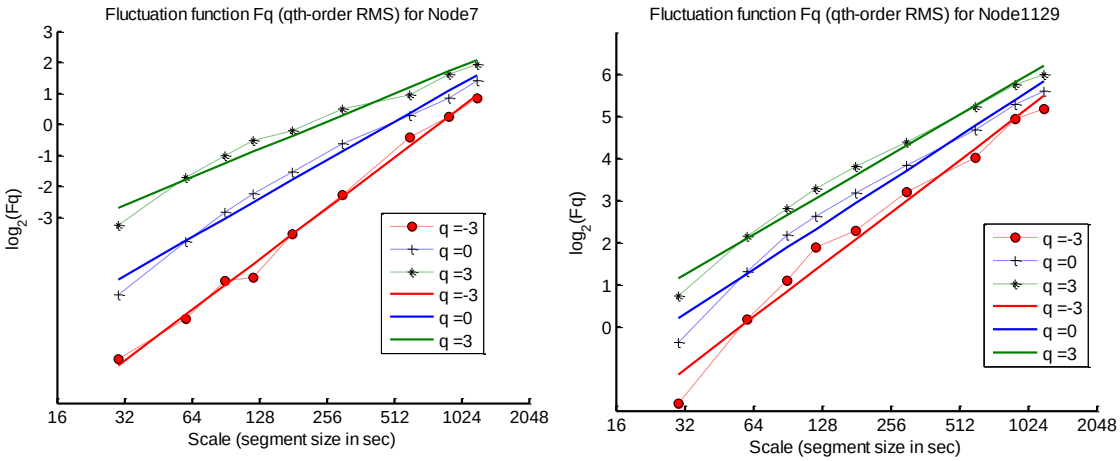
Next the q-order statistical moments (q = -3,-2,-1, 0, 1, 2, 3) are computed for the local RMS values over all the time segments to obtain an overall qth-order RMS or fluctuation function F_q. The q-order weights the influence of segments with large and small fluctuations. The negative q-order (q = -3, -2, -1) amplifies the segments with extremely small RMS values, whereas positive q-order (q = 3, 2, 1) amplifies the segments with extremely large RMS values. The fluctuation function should be computed for multiple scales to emphasize both the fast and slow evolving fluctuations that influence the structure of the time series. Therefore, the scaling behaviour of the fluctuation function is analysed by log-log plots of F_q versus the time scale s for each value of q. Figure 4-17 shows the log-log plots of F_q versus the time scale s for each value of q as dashed lines, and the corresponding regression lines as bold lines.

The existence of long range correlations is confirmed by the existence of a power law relationship between F_q computed for multiple scales as follows [35],[36]:

$$F_q(s) = s^{H_q} \tag{4-5}$$

where, q-order Hurst exponent, H_q is the slope of the regression line, for each qth-order RMS. It can be seen from Figure 4-17(a)-(b) that the difference between the qth-order RMS for the positive and negative values of q is visually more apparent at the smaller scales compared to the larger scales. The small scales are able to

distinguish between the segments with high and low fluctuations (i.e. positive and negative values of q, respectively,) because the small scales are embedded within these segments. In contrast, the large scales cross several segments with both high and low fluctuations and will therefore average out their amplitudes.



(a) (b)

Figure 4-17 qth-order RMS and corresponding regression lines for multiple scales for (a) Cyprus (b) India dataset

It can be observed from Figure 4-18(a)-(b) that H_q is dependent on q indicating that the time series is multi-fractal in nature. The Hurst exponent, $H_{q=2}$ value is close to 0.89 for the Cyprus dataset as shown in Figure 4-18(a) and for the Indian dataset; Hurst exponent $H_{q=2}$ value is 0.9 as shown in Figure 4-18(b) which confirms that the pollution time series have long range correlations.

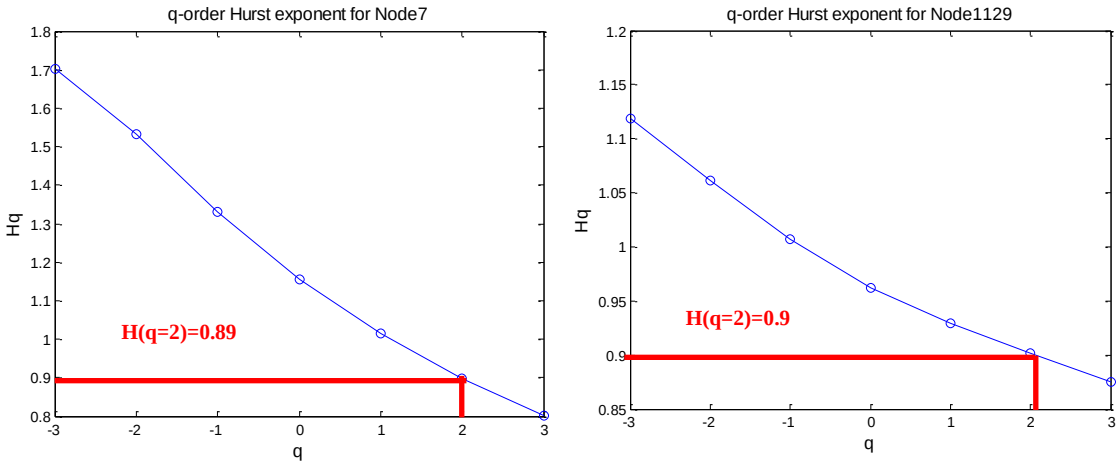


Figure 4-18 Variation of q-order Hurst exponent vs. q for (a) Cyprus (b) India dataset

Next the mass exponent t_q is computed from H_q as follows:

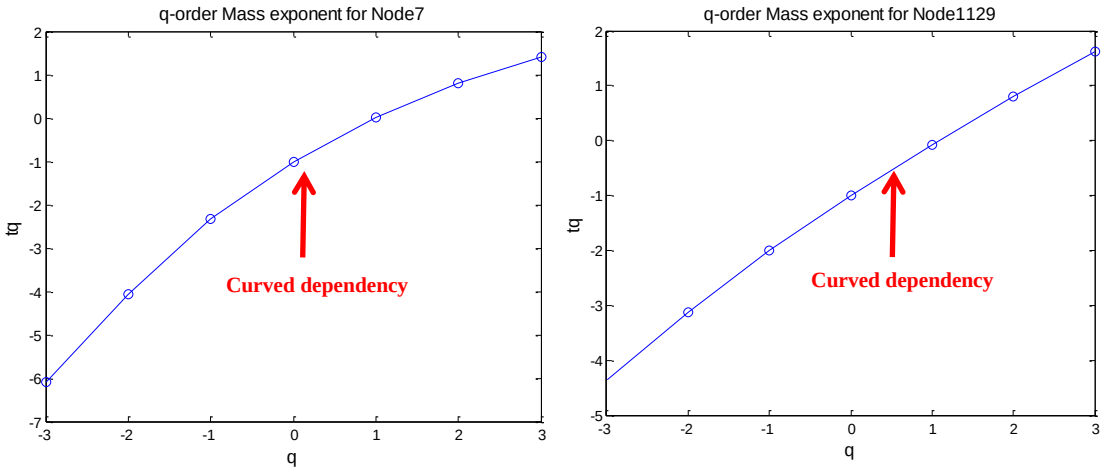
$$t_q = qH_q - 1 \tag{4-6}$$

The mass exponent t_q is used to compute the *q-order singularity exponent* h_q and the *q-order dimension* D_q, [35],[36] as follows :

$$h_q = \frac{d}{dq}[t_q] \tag{4-7}$$

$$t_q = q(h_q) - D_q \tag{4-8}$$

The mass exponent t_q is plotted in Figure 4-19(a)-(b) and it can be seen that the mass exponent t_q has a curved q-dependency which is the characteristic of a multi-fractal time series.

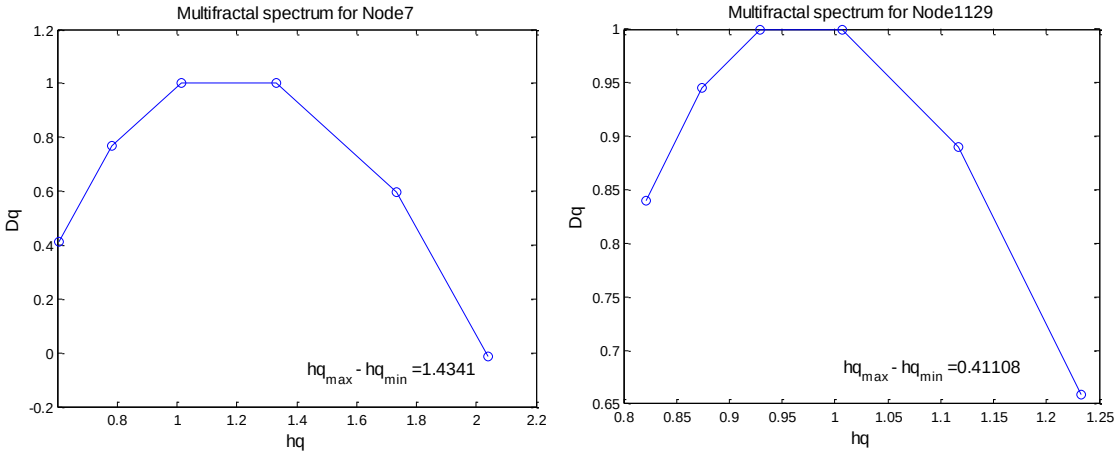


(a) (b)

Figure 4-19 Variation of q-order mass exponent tq for (a) Cyprus (b) India dataset

Next the multi-fractal spectrum, which is the plot of the q-order dimension D_q versus the q-order singularity exponent h_q, is shown in Figure 4-20(a)-(b). The resulting multi-fractal spectrum is a large arc where the difference between the maximum and minimum h_q indicates the multi-fractal spectrum width. The presence of multi-fractal characteristics confirms the presence of non-linear characteristics in the pollution datasets. It can be seen that the multi-fractal spectrum for both the datasets have a long right tail. The long right tail indicates that the multi-fractal

structure is insensitive to the local fluctuations with large magnitudes. Also, the multi-fractal spectrum width for the India dataset is lower than the Cyprus dataset and it can be attributed to the different structure of the time series.



(a) (b)

Figure 4-20 Multi-fractal spectrum for (a) Cyprus (b) India dataset

As mentioned, the multi-fractal de-trended fluctuation analysis (MF-DFA) [35],[36] technique can be used to examine the local fluctuations of the data produced by a node at a given time scale. Because of the existence of the spatial correlations, the neighbouring nodes would exhibit similar fluctuations in the corresponding time segments. This spatial correlation is illustrated for both the Cyprus and the India datasets in Figure 4-21 and Figure 4-22. A fifteen minute (900s) time scale is used to show the local fluctuations for the various sensor nodes.

Figure 4-21 and Figure 4-22 show that the first order RMS values for the various spatially co-located nodes have similar fluctuations in a given time scale. These spatial correlations can be exploited to design the adaptive sampling technique in which the adjacent located nodes can refrain from sampling redundant data. If the representative node is able to capture the event, the neighbouring nodes can avoid capturing the same event. This insight has motivated the design of the spatial sampling algorithms in Chapters 6 and 7.

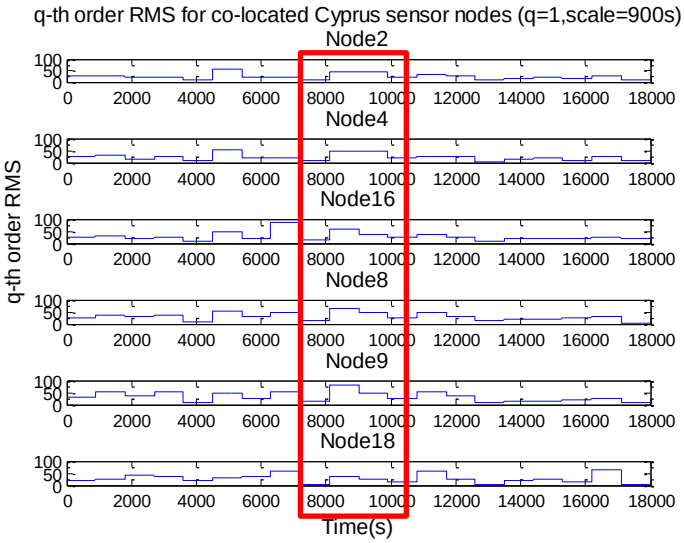


Figure 4-21 Local fluctuations for spatially co-located Cyprus nodes (time scale=900s)

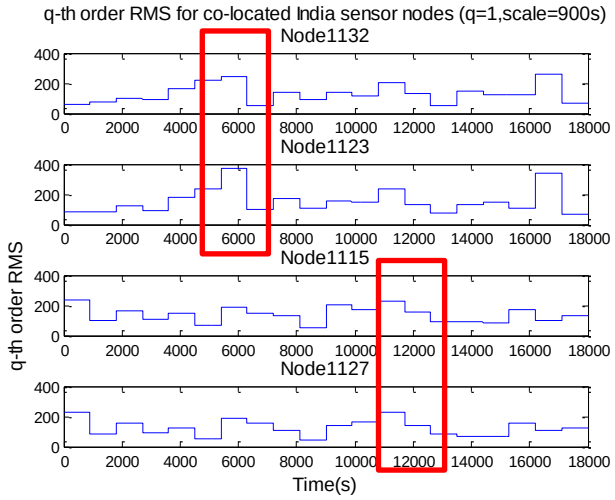


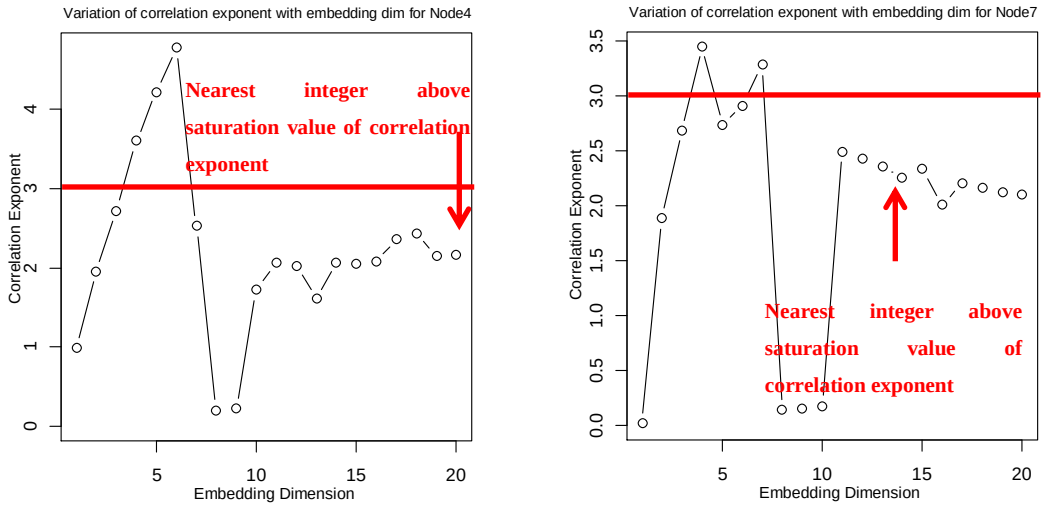
Figure 4-22 Local fluctuations for spatially co-located India nodes (time scale=900s)

4.5 Selection of time delay embedding parameters

Pollution data exhibits non-linear dynamics as was explained in Section 3.6 and time delay embedding [48] is the first step in the construction of the phase space representation of a non-linear dynamical system. Embedding dimension m and embedding delay τ need to be selected in order to create a proper phase space representation. The correlation dimension [48] method as described in Section 3.6 is used to estimate the embedding dimension for the pollution datasets and the results are presented for the Cyprus and the India data here. The idea is to construct a function, *correlation integral* $C(r)$ that is the probability that two arbitrary points on

the orbit are closer together than r. This is usually done by calculating the separation between every pair of N data points and sorting them into bins of width proportional to r. The *correlation exponent* is the slope of $log\ C(r)$ versus $log\ r$ plot. In this case, the correlation exponent for each node is derived using an hour of data. For each of the nodes, the correlation exponent is plotted versus the embedding dimension. The saturation value of the correlation exponent is defined as the *correlation dimension* of the attractor, and the nearest integer above the saturation value provides the minimum number of the embedding dimensions of the phase-space required to model the dynamics of the attractor.

It can be observed from Figure 4-23 that the minimum number of variables to model the dynamics of the CO process for the Cyprus data is 3, because it is the nearest integer above the saturation value of the correlation exponent and it provides the number of dominant variables influencing the dynamics of the underlying system. Also, as the embedding dimension increases, the correlation exponents reach a saturation value, which indicates that the process generating the time series is not random but deterministic. This characteristic is present across different nodes as Figure 4-23 presents (for node 4 and 7).



(a) (b)

Figure 4-23 Correlation exponent and embedding dimension for Cyprus datasets

Similarly the correlation dimension method is applied to the data from two different nodes of the Indian trial as shown in Figure 4-24. The embedding dimension obtained for both the day and the night data is 3. The embedding dimension

parameter estimated from these datasets is used to obtain the time delay embedding for the spatial sampling algorithm in this work and more details can be found in Chapter 7.

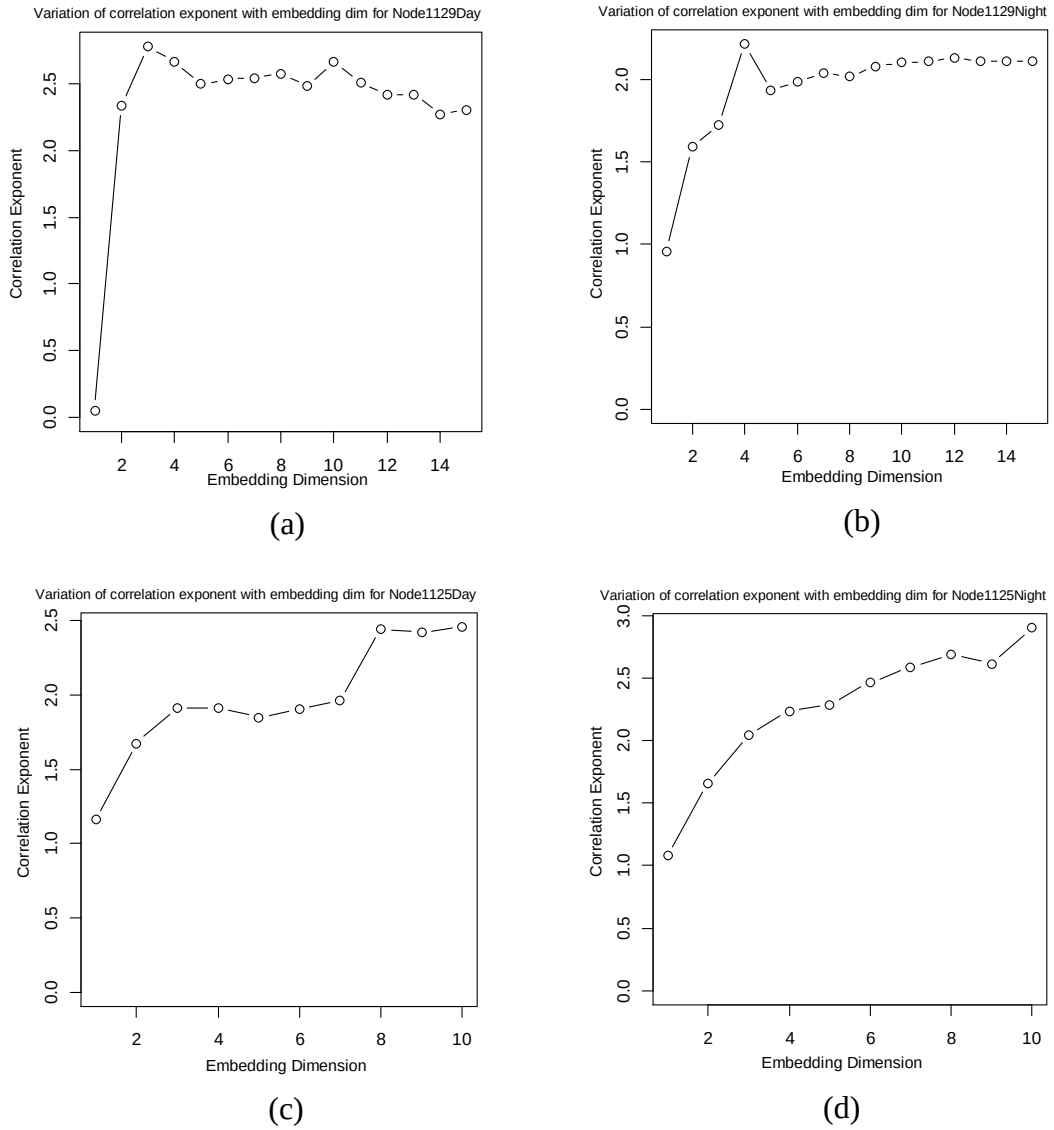


Figure 4-24 Correlation exponent and embedding dimension for India day and night time datasets

Embedding delay values can be estimated using the method of autocorrelation function and mutual information function as explained in Section 3.6. According to the method of autocorrelation function, the choice of time delay is the point where the autocorrelation function drops exponentially to 1/e or 0. The mutual information method provides an information theoretic analogue to the autocorrelation function and the time delay value selected is at the minimum of the average mutual information. Therefore the embedding delay is investigated using three different

methods: first zero of ACF, denoted as *acfzero*; 1/e of ACF, denoted as *acfdecor;* and time delayed mutual information, denoted as *Mutual.* The methods are applied to various pollution time series and the results are shown in Table 4-2. The results shown are calculated over twenty minutes for each time series.

Table 4-2 Embedding delay values using different methods

Cyprus node id	**acfzero**	**acfdecor**	**Mutual**
16	116	32	43
7	79	26	112
4	119	44	34
India node id			
1129	108	32	31
1122	93	60	23
1119	33	22	21

Table 4-2 shows that each method produces different results. Though in this work, the embedding delay is chosen on the basis of the *acfdecor* method, since this is the most popular method suggested in the research literature. According to the *acfdecor* method, an intuition to set the embedding delay values to lower values (20s to 40s) is obtained.

4.6 Chapter conclusions

This chapter has provided details about the pollution trials and the CO datasets collected thereof. The collected datasets have been used in this research work and different data characteristics have been studied using techniques from time series analysis. Different techniques like trend analysis, autocorrelation analysis, ARIMA/EDS forecasting, multi-fractal analysis and selection of time delay embedding parameter have been carried out. The results from the application of these time series concepts to both the Cyprus and the India datasets have been presented in details. The main insights drawn from the data analysis is that the fine grained pollution data possess locally linear trend and slowly decaying autocorrelation. They also exhibit long range dependence and this leads to the existence of self-similarity. Further, they also possess multi-fractal characteristics and non-linear dynamics. Based on the presence of these underlying data characteristics, novel adaptive sampling algorithms will be proposed in the forthcoming chapters 5 and 7.

Chapter 5
Exponential double smoothing based adaptive sampling

5.1 Introduction

As described in the previous Chapter 2 and Chapter 4, applying an efficient temporal adaptive sampling technique to pollution data time series can lead to substantial sampled data reduction and thereby, yield higher sensor energy savings. However, it must ensure that the important changes/events happening in the environment are not lost. In this chapter, a novel design of a temporal adaptive sampling technique called *Exponential Double Smoothing based Adaptive Sampling* (EDSAS) is proposed. The algorithm exploits the existence of local linear trends and slowly decaying autocorrelations as proven in sections 4.3.2 and 4.3.3. The exponential double smoothing (EDS) forecasting [32] was introduced in Section 3.3.1 and it uses the current data and trend level for generating the data forecasts. The algorithm design uses a variant of the EDS forecasting for irregularly sampled time series and avoids the sampling of redundant data points. The main advantages of the proposed technique are the design simplicity and the flexibility, low computational and memory requirements. The technique is evaluated using real pollution datasets and the performance results for various metrics, for example, sampled data reduction and data accuracy are presented. The algorithm performance is compared against another random walk model based stochastic sampling technique called e-Sense, [57],[58] which requires offline data training for model construction to obtain good sampling performance.

The detailed layout of the chapter is as follows: Section 5.2 gives a literature survey on various adaptive temporal sampling techniques for WSNs and points out the research gap that the current work tries to address. Section 5.3 explains the main algorithm principles with background theory on the Wrights extension of the exponential double smoothing technique and also specifies the detailed algorithmic

design. Section 5.4 presents the different performance metrics used for the performance evaluations and the results obtained for EDSAS using different algorithm parameters. It also presents the performance comparison of EDSAS against the e-Sense algorithm (both self-trained and cross-trained against other pollution datasets). Section 5.5 describes the application of EDSAS to temperature and humidity datasets. Section 5.6 draws the conclusions of the chapter.

5.2 Energy efficient sampling in the temporal domain

As mentioned in Section 2.2.2, a detailed survey for different energy management strategies for power-hungry sensors can be found in [5],[6]. The survey includes taxonomy of techniques including hierarchical sensing; adaptive sampling and model based sensing for energy efficient data acquisition. Adaptive sampling techniques dynamically adapt the sampling rate by exploiting the correlations among the sensed data. Model-based sampling techniques build a model of the sensed phenomenon and predict the next data using the model instead of sampling the quantity of interest, hence saving the energy consumed for data sensing. Other than the above energy saving data acquisition techniques, there are data driven energy conservation approaches like the data reduction techniques present in the research literature. Data reduction techniques [6] are different from adaptive sampling techniques. The aim of data reduction techniques is to reduce the amount of data to be delivered to the sink, while adaptive sampling is about reducing the amount of sampling independently of the data reduction scheme employed.

Though, these techniques were briefly described in Section 2.2.2, here a survey of the various techniques is done in order to explore the design space and understand the research gaps in the existing literature. In the following sub-sections, a survey of temporal sampling algorithms/techniques under the categories of adaptive sampling, model based sensing and data reduction that are found to be most relevant to the work at hand is presented.

5.2.1 Survey of adaptive sampling techniques

First, a survey of adaptive sampling techniques is provided. One of the algorithms for adaptive sampling proposed by C. Alippi et al. in [53] is based on the use of

cumulative sum (CUSUM) change detection technique [54]. It dynamically estimates the current maximum frequency of a signal and indicates a change when the current maximum frequency is above or below a threshold for some number of consecutive readings; this triggers an update to the sampling rate. The computational load of the proposed algorithm is quite high and a centralised approach is taken. The update algorithm is executed at the base station and the resulting sampling rates are sent to each sensor node. Hence, the approach is not very scalable and incurs significant communication overhead in the network.

A second approach for adaptive sampling is a statistical protocol proposed by Jain A., Chang E. Y in [55] in which each node adapts to the streaming data characteristics. According to this method, the nodes autonomously decide their sampling rate within a given range using a Kalman Filter (KF) to estimate the error. When the desired sampling rate violates the range, a new sampling rate is requested to the base station. The base station then determines new sampling rates, taking into account the constraints of available resources, such that the KF estimated error over all active nodes is minimised. In this approach, the sensors transmit their sensed data to the base station. This is, again, neither scalable, nor is it suitable for distributed implementation because of the computational complexity of the KF [56].

Yet another sampling technique is applied for a flood warning system called Floodnet [4]. It uses a centralized flood predictor model that is used to determine the priorities for collecting samples from each sensor. The model comprises a stochastic one dimensional numerical hydraulic model coupled to an ensemble Kalman filter. The model allows real time collection of water depth data to update the flood predictions regularly with refreshed water levels. When the model based probability for the water level exceeding a threshold is less than 5%, the requirement for data transmission from a node is lowered, otherwise, the requirement for data transmission is raised. The degree of data transmission is related to the importance factor, i.e. data with high sampling rates are more important and are associated with critical zones (variation of phenomenal dynamics is higher). Upon each iteration of the model, the network changes its behaviour, altering the reporting rate (derived from the data importance using a conversion function specified by the environmental experts) of each individual node according to the data importance placed by the predictor model.

As it can be seen all the above mentioned adaptive sampling techniques are centralized in nature and require great amount of computation, hence simpler, decentralized adaptive sampling techniques are required.

5.2.2 Survey of model based sampling techniques

Second, a survey of model based techniques is presented. One of the model-based adaptive sensing algorithms is proposed by H. Liu et al. called e-Sense in [57]. This is a stochastic sampling scheduler that uses a biased random walk approach [58] to predict the likelihood of an event happening k time steps into the future and based on the calculated probability, it determines a sampling probability for that instant. The stochastic scheduling problem is formulated as an optimization problem that minimizes the total energy consumption while providing statistical guarantees on data sampling quality.

e-Sense is largely based on the assumption that the environmental data follow a random walk model and because of the generic and simple nature of the model, this algorithm is chosen as the reference for comparison purposes with the proposed temporal sampling algorithm in this research work. More detailed description about e-Sense can be found in Section 5.4.2.1. The e-Sense technique has the prerequisite of an accurate data model. An inappropriate choice of a model will affect the performance of the algorithm. In addition, the re-training of the e-Sense data model can only be done on a node with sufficient memory and processing power.

Barbie-Q- a tiny-model query system (BBQ)[59] is a data acquisition model for sensor networks that incorporates statistical models of real-world processes into a sensor network query processing architecture. BBQ tries to identify a data acquisition plan for the sensor network and provide the best query answer, given a query and a model. BBQ uses a specific model based on time-varying multivariate Gaussians in its architecture to compare it with the incoming data stream. If all the probabilities generated meet or exceed a user specified confidence threshold, then the requested readings are directly reported as the means in the probability density function. BBQ imparts confidence on the posterior density generated by time varying multivariate Gaussians and optimizes the expected benefit and cost of observing the attributes. BBQ depends heavily on the prediction ability of the time varying multivariate Gaussians. In case of high nonlinearity among the sensed observations, the predictive

ability of the multivariate Gaussians fails, thereby responding to the query incorrectly, despite having a confidence in the model due to the previously sensed observations.

Another algorithm that exploits temporal correlations between sensed data is the Utility Based Sensing and Communication (USAC) [60] algorithm. This is a decentralized technique for adaptive sampling, designed for an environmental WSN measuring sub-glacial movement in the Glacsweb project. The adaptive sampling aspect of the algorithm models temporal variations in the environmental parameter being sensed as a piecewise linear function, and then uses a pre-specified confidence interval parameter in order to make real-time decisions regarding the sampling rate of the sensor nodes. Linear regression is used to predict the value of future measurements, and if the actual sensor reading exceeds the confidence interval parameter, the sensor starts sampling at an increased rate. One of the drawbacks of this technique is that the sensor can rapidly deplete its battery if the increased sampling rate is constantly retriggered by data that is not linear.

5.2.3 Survey of data reduction techniques

Third, a survey of data reduction techniques is presented. There are time series forecasting based data reduction techniques like Probabilistic Adaptable Query system (PAQ) [61], Similarity-based Adaptive Framework (SAF) [62]that use dual prediction. They work by collecting readings, comparing the readings with the forecasts, and if the forecasts differ enough from the readings, updating the model. PAQ [61] is based on a low-order Auto-Regressive (AR) model, with the aim of reducing the amount of computation to be performed by sensors. The first instance of the model is computed by sensor nodes using a set of sampled values. During this learning phase, nodes store the samples in a queue. When the queue is full they can get the model and send it to the sink. The communication between nodes and the sink is limited to the parameters of the model (i.e. the coefficients of the AR model), and it does not include the sensor readings. Each model is associated to a user-specified error bound. When a predicted value falls within the error bound, the model is considered valid for the given sensed quantity. It also proposes a distributed clustering scheme to group the similar sensor nodes and represent them by the same model within a given user specified threshold. SAF

[62] refines the AR model used in the previous work to include a trend component and obtain better prediction of the phenomena with sharp variations in their values.

Liu et al. [63] uses the idea similar to PAQ and SAF to keep sensor nodes from transmitting redundant information, which can be predicted by the sink node. An auto-regressive integrated moving average (ARIMA) model is used as the data modelling method, and energy efficiency is achieved by suppressing the transmission of samples, whose prediction values based on the ARIMA model are within a pre-defined tolerance value from their actual values.

It can be noted that all the above mentioned data reduction techniques use AR models. The AR models have certain disadvantages that have been listed in section 3.3.3 and also the forecasting capabilities are inferior in comparison to exponential double smoothing based forecasting for pollution data as shown in Section 4.3.4.

5.2.4 Conclusions from the literature survey

The key observations that can be made from the literature survey carried out in previous sections are as follows:

1) Most of the temporal adaptive sampling techniques that exist in the literature are centralized in nature and computationally intensive.

2) The model based adaptive sampling techniques do not take into account the particular data characteristics for the application at hand, for example, pollution monitoring in the current work. They typically are based on measurements from sensors that measure physical processes that are fundamentally different to that of pollution (for example, temperature).

3) Although there are data reduction schemes that use lower order AR models for data prediction, there is a gap in the literature where alternative time series forecasting methods like exponential double smoothing, is applied for adaptive sampling in WSNs.

Therefore, the goal in this chapter is to develop a light weight, real-time technique for adaptive sampling using time series forecasting methods that take into account the data characteristics like trends and autocorrelations to provide the best trade-off between sampled data reduction and data accuracy.

5.3 Algorithm design and details

Pollution datasets exhibit local linear trends and slowly decaying temporal correlations as shown in sections 4.3.2 and 4.3.3. Therefore, a time series forecasting can be used to predict only the most significant data points and avoid sampling the unnecessary data points. An introduction to exponential double smoothing (EDS) was given in Section 3.3.1 that uses simple recursive equations to generate forecasts based on the current level and trend in the time series. One of the drawbacks of EDS forecasting is that it requires the data to be sampled at regular time intervals. In case of pollution data, due to the slowly decaying autocorrelations, data may stay almost the same for consecutive time intervals and need not be sampled. Consequently, the requirement to sample the data at irregular time intervals arises, i.e. only when the data undergoes a significant change. In order to address forecasting at irregular time intervals, an extension to EDS called the Wright's extension [64] is proposed to be used for the pollution datasets. An adjustment feedback loop that uses an exponentially weighted moving average (EWMA) [66] based change detection mechanism, as introduced in Section 3.3.1, has also been included in the proposed algorithm design. The novel algorithm is termed as Exponential Double Smoothing based Adaptive Sampling (EDSAS). The advantages of EDSAS are as follows:

1) There is no need for offline training phase for data model construction. The prediction can be carried out on individual sensor nodes.

2) The data does not need to be sent to a base station for model construction. Consequently, EDSAS is decentralized in nature and has lower communication requirements.

3) The computational and memory requirements of EDSAS are low, as little historical data is required for the calculation of EWMA and EDS prediction using simple, recursive mathematical equations. Taken together, these features mean that EDSAS processing can be carried out on most CPU and memory constrained sensor nodes.

In the rest of the section, the main mathematical concepts behind Wright's extension to Holt's method are explained in order to understand the detailed algorithm design.

5.3.1 Wright's extension to Holt's method

EDSAS is based on EDS, which is an exponential smoothing method for time series forecasting as described in Section 3.3.1. Although the double-parametric form of EDS can be used for datasets with linear trends, it assumes regular data sampling intervals. A modification of EDS using Wright's extension that works for irregularly sampled time series is adopted in the algorithm design. According to the Wright's extension, the next k-step forecast is obtained by using the following relationships [64],[65]:

$$\hat{y}_{t+k_t} = L_t + k_t M_t \tag{5-1}$$

where, t is an index of the time series and $t = 1,2,\ldots.n$; L_t and M_t represent the estimate and trend respectively at interval t; and k_t is the step size between the observations at interval $t+1$ and t. The step size k_t can vary dynamically and take different values across the two time instants $t+1$ and t. This enables Wright's extension to forecast for irregularly sampled time series. The update equations for L_t and M_t follow the basic principle as follows:

$$\begin{aligned} &new\ estimate \\ &\quad = (parameter)new\ information \\ &\quad + (1 - parameter)previous\ estimate \end{aligned} \tag{5-2}$$

and are given by the following relationships [64],[65]:

$$L_t = V_t y_t + (1 - V_t)(L_{t-1} + k_{t-1} M_{t-1}) \tag{5-3}$$

$$M_t = U_t(L_t - L_{t-1})/k_{t-1} + (1 - U_t)M_{t-1} \tag{5-4}$$

The previous estimate for L_t is $L_{t-1} + k_{t-1}M_{t-1}$ and the new information about L_t is the actual data reading y_t. The previous estimate of M_t is M_{t-1} and the new information about M_t is $L_t - L_{t-1}/k_{t-1}$. Also, in equations (5-3) and (5-4), the parameters V_t and U_t are changed to reflect the changes in time spacing as follows [64],[65]:

$$V_t = \frac{V_{t-1}}{b_t + V_{t-1}} \tag{5-5}$$

$$b_t = (1-\alpha)^{k_{t-1}} \quad (5\text{-}6)$$

and

$$U_t = \frac{U_{t-1}}{d_t + U_{t-1}} \quad (5\text{-}7)$$

$$d_t = (1-\beta)^{k_{t-1}} \quad (5\text{-}8)$$

where, V_t and U_t are normalising factors and α, β are the smoothing parameters such that $0 < \alpha < 1,\ 0 < \beta < 1$. All the variables L_t, M_t, V_t, U_t need to be initialized appropriately. L_t and M_t are set according to the initial level and trend values of the time series. V_t and U_t are initialized using the following relationships [64] where q denotes the average time spacing for the data [64]:

$$U_t = 1 - (1-\beta)^q \quad (5\text{-}9)$$

$$V_t = 1 - (1-\alpha)^q \quad (5\text{-}10)$$

The average time spacing for the pollution datasets used in this work is 1s. The following section explains the EDSAS detailed design and how theoretical concepts explained here are applied for adaptive sampling.

5.3.2 EDSAS technique

The proposed EDSAS algorithm operates in two stages. The block diagram of EDSAS is shown in Figure 5-1. The first stage is used for step size calculation based upon the forecast error between the forecasted and actual data values. A step size is defined as the time interval between two consecutive sampled data points. The step size is used in the time series prediction using Wright's Holt's extension to generate a data forecast for the next sampling instant. The second stage is used to carry out the feedback loop for step size adjustment based on EWMA. If the EWMA based change detection finds any important event/change happening in the environment, the step size is brought back down to sample the data at shorter time intervals.

In the stage 1, at every sampling instant, a forecast accuracy measure, F_e, is computed between the actual sampled data value and the forecasted data value computed at the previous sampling instant. The forecast accuracy measure, F_e provides an estimate of the forecast error. Depending upon the forecast error, the next

sampling instant is calculated by outputting a parameter, k, i.e. the *step size.* The step size parameter k is initialised to 1 and can be incremented to a user defined maximum step size S_{max}, which is determined by the application data accuracy requirements. At every sampling instant, the step size is successively increased by one, if the forecast accuracy measure F_e stays below a user defined *error tolerance level*, δ. This is repeated until the step size reaches a *maximum step size*, S_{max}, otherwise the step size is reduced by one. The step size value obtained is used to perform the k-step time series data prediction using the equations of the Wright's extension to Holt's method from Section 5.3.1 and the forecasted values are used at the next sampling instant.

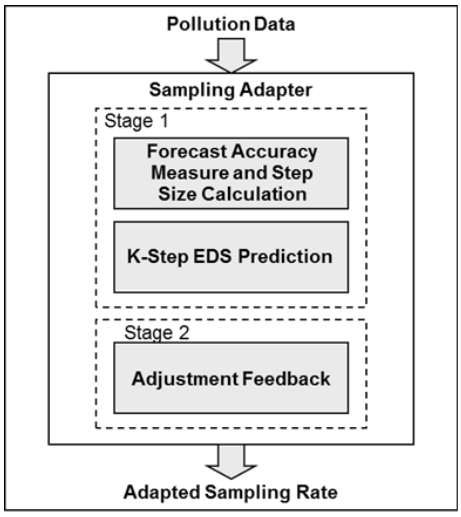


Figure 5-1 Block diagram of EDSAS

The stage 2 comprises of an adjustment feedback, which is triggered when the sampling rate reaches the maximum step size, S_{max}. Once S_{max}, is reached, the data will continually be sampled at S_{max} intervals provided that the forecast error, F_e stays below the error tolerance, δ. The adjustment feedback is necessary to maintain data fidelity and minimize the number of false misses. False misses are defined as the important changes/events happening in the environment, which could not be detected during the maximum step size sampling intervals. In a way, the false miss ratio provides an indication of the guarantees on data accuracy provided by the sampling technique and will be used as one of the metrics in evaluating the algorithm performance. The adjustment feedback uses exponentially weighted moving averages (EWMA) as the basis of a change detection mechanism as explained in Section 3.3.1. The ratio η of the short term moving average L_{short} and the long term moving

average L_{long} is computed using two smoothing parameters, α_{long} and α_{short} respectively. If η exceeds a threshold, a rare event occurrence can be detected and used to bring down the value of the step size, k.

The pseudo code for the EDSAS algorithm is as follows:

Algorithm 5.1 EDSAS adaptive temporal sampling

```
Initialize EDSAS variables k←1, δ, S_max
Initialize EDS smoothing parameters α, β
Initialize EWMA smoothing parameters α_long, α_short
Initialize Wright's extension variables L_t ← y_t, M_t ← y_{t+1} - y_t, V_t ← 1 − power((1 − α), k), U_t ← 1 − power((1 − β), k) ,t ← 1
Initialize forecast ŷ_{t+k} ← L + k ∗ M

function senseData(t+k)
    while (dataCollection == TRUE)
        y_{t+k} ← collect data sample
        adaptiveSampling(y_{t+k}, ŷ_{t+k}, k, δ, S_max)
    end while
end function

function adaptiveSampling(y_t, ŷ_t, k, δ, S_max)
    Calculate the forecast accuracy measure
    if (y_t - ŷ_t < δ) then
        Perform step size modification
        if (k < S_max) then
            increment k
        end if
    Else
            decrement k
    end if
    ŷ_{t+k} ← perform k-step prediction using modified EDS equations
    if(k == S_max) then
        if(consecutive S_max predictions) then
            update L_short, L_long
            Perform the change detection
            evaluate η ← L_short / L_long
            Carry out the feedback adjustment
            if ( η > 1) then
              k ← 1
            end if
        Else
            reset L_long, L_short
        end if
    end if
    senseData(t+ k)
end function
```

Figure 5-2 shows the process of modification of the step size in EDSAS. The step size constantly keeps increasing until it reaches S_{max} and stays at S_{max} as long as no

change is detected. If any change is detected the step size is brought down to one, i.e. k is set to one.

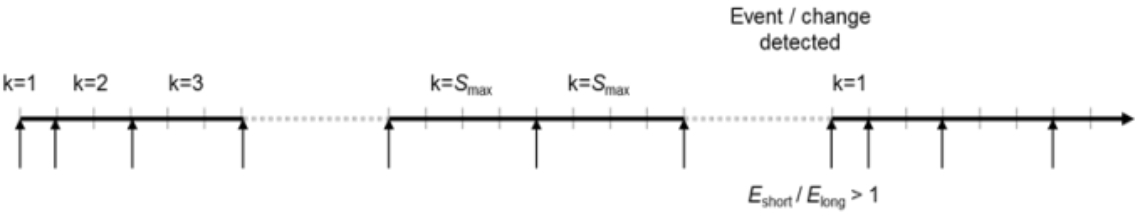


Figure 5-2 Step size modification in EDSAS

All the algorithm input parameters are initialized to appropriate values depending on the application requirements and while the data collection process is in progress, EDSAS uses the Wright's extension to predict the future data points. If the forecast error, F_e is less than the error tolerance, δ, then the step size is increased and the sensor node can avoid sampling unnecessary data points for the computed step size interval. On the other hand, if the forecast error exceeds the tolerance limits or dynamic changes are happening in the environment, then EDSAS will decrease the step size and revert back to sampling at shorter step size intervals. Hence, it can be seen that EDSAS technique is simple in operation and it can be easily executed on resource constrained devices. The algorithm can adapt dynamically to capture the changes happening in the environment, provided that the algorithm parameters are chosen appropriately to meet the application requirements.

In the following Section 5.4, the detailed EDSAS performance analysis is presented.

5.4 Performance evaluation of EDSAS

Different performance metrics are used to evaluate the sampled data reduction and data accuracy levels yielded by EDSAS. The details about the different metrics selected for evaluating the performance of EDSAS are as follows:

1. *Sampling fraction* (SF): This defines the ratio of the total number of data samples taken by the sampling algorithm versus the total number of data points available in the real dataset. Smaller values of sampling fraction (less number of sampled data points) correspond to higher sensor energy savings achieved by the sampling algorithm.

2. *Miss Ratio* (MR): This is the fraction of the number of events/changes that have not been detected (false misses) by the algorithm for a given relative threshold. Relative threshold is defined as the difference between the last and the current measurements that triggers an event. The actual number of changes or "true" changes can be calculated from the real dataset for the given relative threshold. If at a particular time instant, the data is not sampled while an actual change happened, then it is considered as a false miss. Say, n_f denotes the number of misses and n denotes the total number of sampling points, then the MR is computed using the following relationship:

$$MR = \frac{n_f}{n} \tag{5-11}$$

3. *Sampling Performance* (SP): $SF$and MR are closely related to each other. In general, MR decreases if SF increases. Therefore, to enable the comparison of different sampling strategies and algorithms, a ratio of these two metrics is considered to compute the sampling performance of the algorithm as follows:

$$SP = \frac{1 - SF}{MR} \tag{5-12}$$

where $1 - SF$ denotes the sampled data reduction obtained by the algorithm and MR denotes the miss ratio. Lower values of SF and MR will lead to higher values of sampling performance, i.e. an algorithm or a selection of parameters that is able to sample less, while not missing important events happening in the environment, shall have a better performance in comparison to another. Therefore, SP provides a performance/cost metric to evaluate the actual benefit of sampled data reduction with the consideration of the reduction in data accuracy.

4. *Mean deviations*: In the atmospheric sciences field, the environmental scientists are generally concerned about the true averages and that is why deviations from the true mean is also used for evaluating the data measurement accuracy as obtained by the sampling algorithm. True means are calculated from the real data, whereas sampled data is used to compute the sampled means. Average percentage deviations between the true means and sampled means are reported across the various nodes.

All the above mentioned performance metrics are evaluated for the various pollution datasets. Performance results for EDSAS are presented in the following Section 5.4.1, while performance comparison against the reference algorithm, e-Sense is presented in Section 5.4.2.

5.4.1 Analysis for different EDSAS parameters

In this section, the effect of the different values for maximum step size (S_{max}) and error tolerance (δ) is evaluated on the sampling fraction, miss ratio and sampling performance metrics of EDSAS. Experiments are carried out using both the Cyprus and Indian pollution datasets. The relative threshold value is set to 0.1 ppm. The basic statistics, i.e. mean, standard deviation, total number of changes for a relative threshold of 0.1ppm for the various Cyprus and Indian datasets are shown in Table 5-1.

Table 5-1 Basic statistics for various (a) India and (b) Cyprus datasets

(a)

Node Id	Mean (ppm)	Std dev. (ppm)	Number of changes
1115	6.52	2.76	19994
1118	3.91	1.69	7722
1119	4.91	2.44	11952
1121	2.77	1.36	10741
1122	2.17	1.05	8475
1123	9.72	3.45	21851
1125	4.43	2.00	7229
1126	1.68	1.29	4987
1127	6.84	2.51	19426
1129	10.70	3.66	17510
1131	12.15	4.65	18269
1132	8.88	2.59	14533
1135	1.76	1.34	8016
1137	4.14	2.02	9384

(b)

Node Id	Mean (ppm)	Std dev. (ppm)	Number of changes
2	0.67	0.43	1460
3	0.79	0.52	1676
4	0.67	0.39	845
5	1.56	1.11	3642
7	1.77	1.54	4297
8	0.68	0.58	845
9	0.85	0.64	1721
11	0.80	0.59	2039
12	0.98	0.64	720
13	0.57	0.41	679
14	0.84	0.59	2299
15	1.76	1.70	4233
16	0.70	0.53	1568
18	0.74	0.54	1548

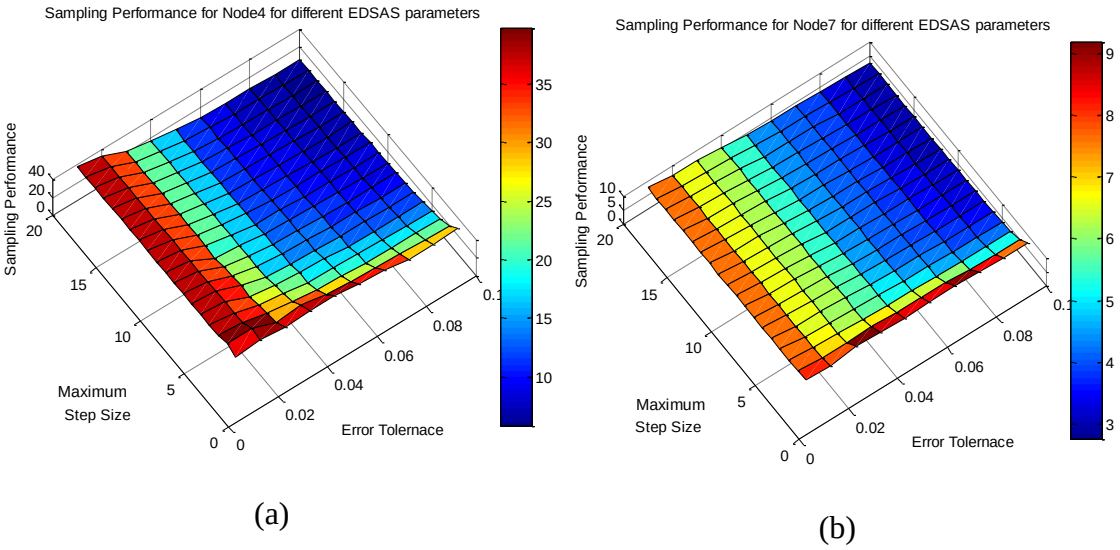
The pilot pollution studies in [12],[13] have shown that the differences in time-averaged CO concentrations at different locations in a street are very small. This is

taken to imply a stringent data accuracy requirement of less than 0.5ppm and is the reason for setting the relative threshold to 0.1ppm. The number of changes shown represents the actual changes happening in the environment and is used to compute the miss ratio for each node.

The smoothing parameter α_{short} is set to 0.9 and α_{long} is set to 0.01 for EWMA change detection. The values for the smoothing parameters α_{short} and α_{long} are chosen so that the most recent data values get the highest weights while the historical data values get the least values. The smoothing parameter α is set to 0.9 and β is set to 0.6 for the Wright's extension for the various experiments carried out with the pollution datasets. The smoothing parameter values α and β are chosen after experimenting with different values ranging from 0.5 to 0.9 and selecting the ones that yield the best sampling performance.

The sampling performance metric is computed for a range of error tolerance and maximum step size values. The error tolerance varies from 0.01 ppm to 0.1 ppm and the maximum step size varies from 1s to 20s. The error tolerance is selected based upon the threshold value for which the changes need to be detected. The maximum step size is selected based upon the autocorrelation analysis in Section 4.3.3 that suggested the autocorrelations are significant at smaller time lags of 10s to 20s.

The results are shown in Figure 5-3(a)-(b) for Cyprus datasets from nodes 4 and 7 out of the fourteen nodes. The colour bar in the surf plot indicates the value of the sampling performance metric as the error tolerance and maximum step size vary for each of the nodes. It can be seen from the surf plots that lower values of S_{max} and δ result in better sampling performance (indicated by red-orange color in the surf plot). The graphs are shown for a few sample nodes (4, 7), but similar behaviour is observed across the rest of the nodes. These experiments provide an insight for the parameter selection as δ being set to 0.07ppm and S_{max} being set to 5 across the various nodes. The individual variation of miss ratio and sampling fractions for these nodes are investigated further for these set of parameters (S_{max} is set to 5 and δ is set to 0.07ppm).



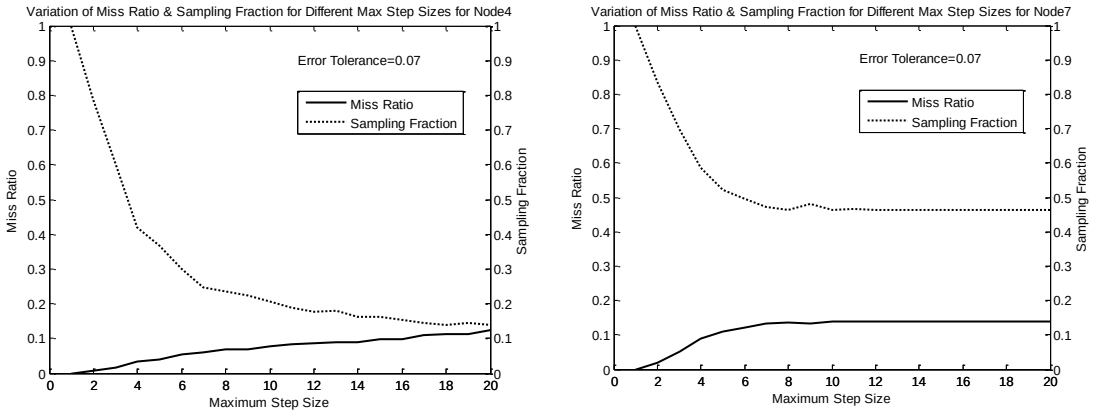
(a) (b)

Figure 5-3 Sampling performance for various parameters for (a) node 4 (b) node 7

Figure 5-4(a)-(b) shows the effect of different maximum step sizes for a fixed error tolerance value of 0.07 ppm. It can be observed from the graphs in Figure 5-4(a)-(b) that MR (shown on the left axis) increases as the maximum step size increases, while SF (shown on the right axis) decreases. It can be further observed that SF for node 4 falls to as low as 0.15 while for node 7, SF falls to 0.45. This implies that the sampled data reduction (1- SF) obtained across these nodes increase as the maximum step size increases to as high as 85% for node 4 and 55% for node 7. MR rises up to a little more than 0.10 for node 4 and almost 0.15 for node 7. The main point to be observed is that the variation in SF and MR is sharper at lower S_{max} values while, the change in SF and MR gradually becomes constant as S_{max} increases. This implies that higher S_{max} values do not have much impact on the sampling performance. The variation in the performance across two nodes 4 and 7 is due to the nature of data measured by them. In case of the node 7, despite sampling more data points, there is higher number of misses due to the more dynamic nature of the data from node 7.

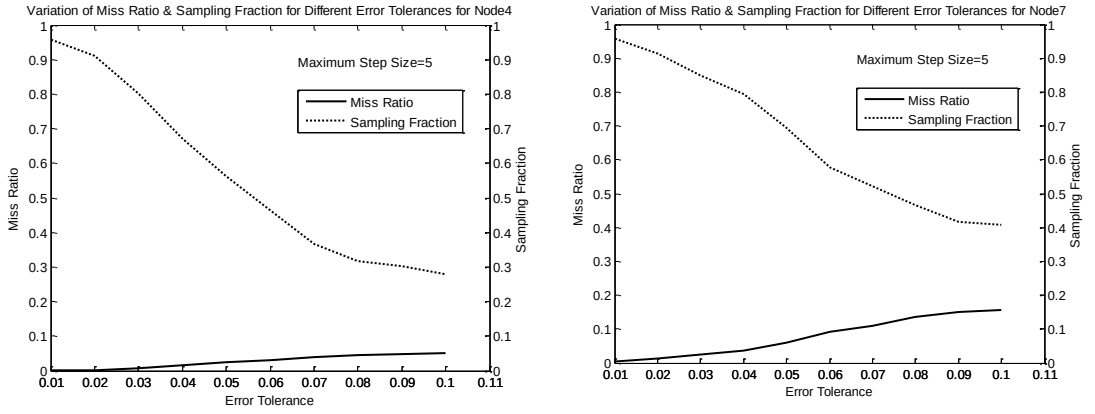
Figure 5-5(a)-(b) shows the effect of different error tolerances for fixed maximum step size value of 5. It can be observed from Figure 5-5(a)-(b) that as the error tolerance increases to 0.1, MR for node 4 stays around 0.05, while MR becomes as high as 0.15 for node 7. SF for node 4 goes up to 0.3, while for node 7, it is around 0.4. It can be observed that as the error tolerance increases, there is a continuous decrease in SF while MR continues to increase. This implies that tuning the error

tolerance has higher impact on sampling performance in comparison to tuning S_{max}. The higher values of error tolerance lead to higher sampled data reduction and consequently, a higher miss ratio.



(a) (b)

Figure 5-4 Variation of MR and SF for different values of S_{max} for (a) node 4 (b) node 7



(a) (b)

Figure 5-5 Variation of MR and SF for different values of δ for (a) node 4 (b) node 7

Finally, EDSAS is run across all the nodes from the Cyprus trial and for this choice of parameters (S_{max} set to 5 and δ set to 0.07ppm), SF varies from 0.32 to 0.52, while MR varies from 0.04 to 0.13 across different nodes. This means sampled data reduction of the order of 48% to 68% can be obtained across all the nodes, while losing 4% to 13% of the changes, as shown in Table 5-2(a). Higher values of S_{max}

and δ can result in further sampled data reduction, but the application data accuracy considerations need to be kept in mind.

Table 5-2 SF, MR and SP for various nodes for (a) Cyprus data (S_{max}=5, δ=0.07) (b) India trials (S_{max}=5, δ=0.08)

(a)

Node Id	Sampling Fraction	Miss Ratio	SP
2	0.39	0.06	10.37
3	0.41	0.07	8.93
4	0.37	0.04	16.64
5	0.48	0.10	5.15
7	0.52	0.11	4.32
8	0.34	0.04	15.12
9	0.39	0.07	9.30
11	0.42	0.08	7.65
12	0.32	0.05	13.00
13	0.36	0.04	18.11
14	0.48	0.07	7.14
15	0.47	0.13	4.23
16	0.43	0.06	9.67
18	0.41	0.06	10.28

(b)

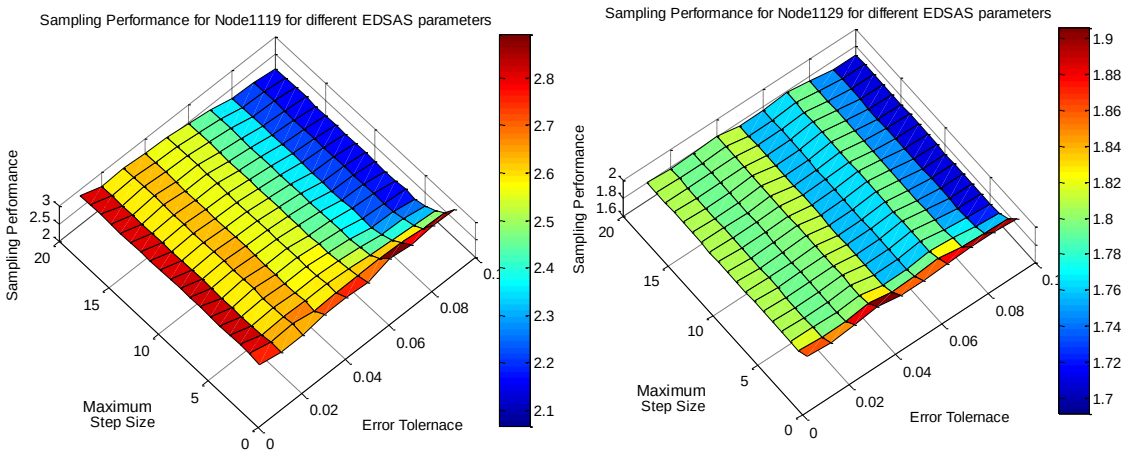
Node Id	Sampling Fraction	Miss Ratio	SP
1115	0.87	0.09	1.56
1118	0.58	0.13	3.13
1119	0.73	0.12	2.30
1121	0.72	0.11	2.55
1122	0.73	0.08	3.39
1123	0.94	0.05	1.38
1125	0.58	0.13	3.18
1126	0.53	0.09	5.06
1127	0.87	0.09	1.51
1129	0.90	0.06	1.75
1131	0.90	0.06	1.67
1132	0.87	0.07	1.93
1135	0.72	0.08	3.53
1137	0.67	0.11	2.91

Similar performance results are generated with the India day time data and the detailed results for sampling performance for India nodes 1119 and 1129 are shown in Figure 5-6(a)-(b). Based on the surf plots, it can be observed that lower values of maximum step sizes and error tolerances give the best sampling performance. The parameter S_{max} is set to 5 and δ is set to 0.08ppm for the various Indian datasets.

The variation of the sampling fraction and the miss ratio for different maximum step size are shown in Figure 5-7(a)-(b). It can be seen that for node 1119, MR is as high as 0.14 for SF of 0.7, while for the node 1129, MR goes upto 0.07 for SF of 0.9. It can be observed from these graphs that at higher values of S_{max}, the variation in SF and MR almost becomes constant. This suggests that S_{max} should be set to lower values and setting S_{max} to higher values does not provide any additional benefit.

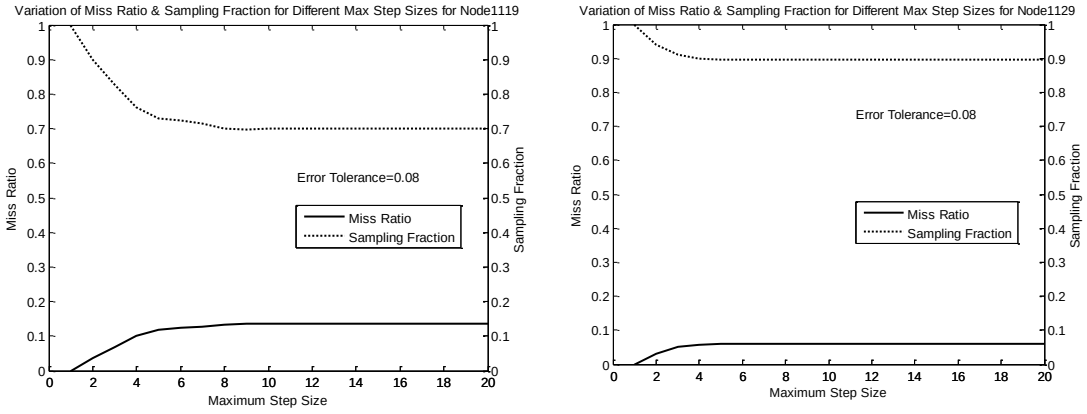
The variation of the sampling fraction and the miss ratio for different error tolerances for the India nodes are shown in Figure 5-8(a)-(b). MR for node 1119 goes to 0.2, while SF is 0.6. SF for node 1129 becomes 0.85, while MR is around 0.08. In this case too, it can be observed that as the error tolerance increases, both the sampled data reduction and miss ratio increase. In case of node 1129, the data reduction does not go beyond 85% because the data is very dynamic and EDSAS executes a tight control by capturing most of the data changes.

It can be seen from Table 5-2(b) that parameter settings of S_{max} set to 5 and δ set to 0.08, results in an overall sampled data reduction of 6% to 47% (SF as high as 0.94 to as low as 0.53) are obtained across various nodes, while losing only 5% to 13% of the changes (MR varies from 0.05 to 0.13). The sampled data reduction obtained for the Indian datasets are lower in comparison to the Cyprus datasets since they are more dynamic in nature and need to be sampled more to capture the changes.



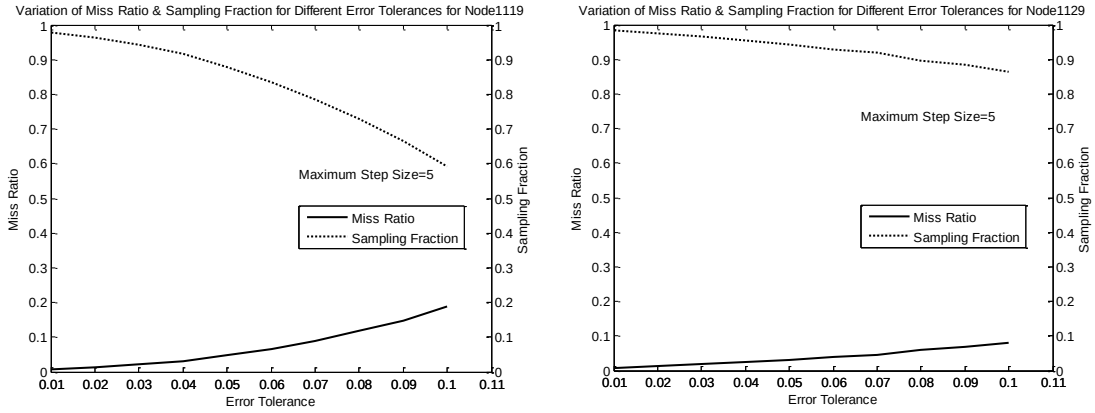
(a) (b)

Figure 5-6 Sampling performance for various parameters for (a) node 1119 (b) node 1129



(a) (b)

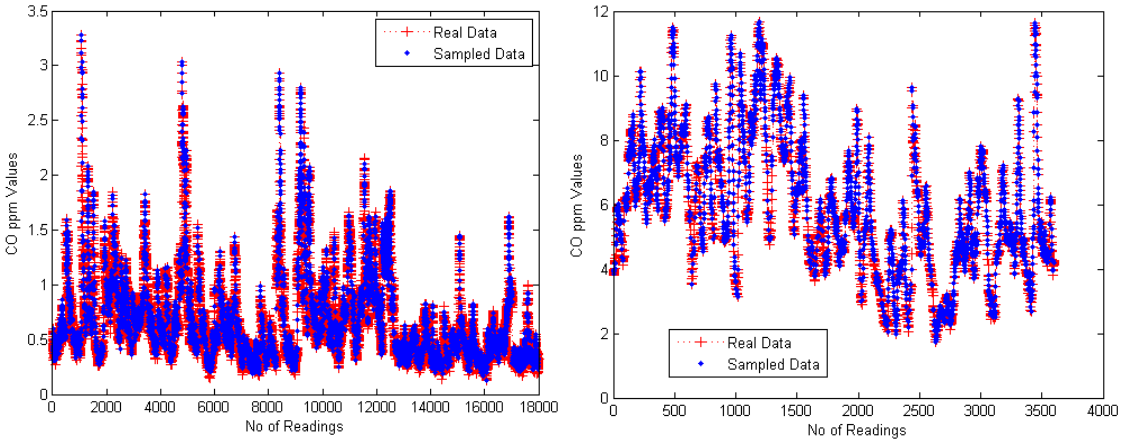
Figure 5-7 Variation of MR and SF for different values of S_{max} for (a) node 1119 (b) node 1129



(a) (b)

Figure 5-8 Variation of MR and SF for different values of δ for (a) node 1119 (b) node 1129

Figure 5-9(a)-(b) shows the original time series and the sampled time series for both the Cyprus (five hour data for Node 4 with S_{max} set to 5 and δ set to 0.07ppm) and India (one hour data for node 1119 with S_{max} set to 5 and δ set to 0.08ppm) datasets. It can be seen that the sampled data follows the original data very closely for this choice of parameters. This shows that EDSAS can be used to reproduce the original data with reasonable accuracy.



(a) (b)

Figure 5-9 Sampled pollution time series for (a) Cyprus (b) India datasets

5.4.2 Performance comparison against e-Sense algorithm

EDSAS is compared against e-Sense [57],[58] that is a model based sampling technique. e-Sense is chosen because it is a data driven, node level sensor energy management technique that aggressively conserves power by reducing sensing activity on the sensor node similar to the objective of EDSAS. It is based on an intuitive random walk based model that is easy to understand and implement. The main aim is to study the implications and performance of a model based sampling scheme for pollution datasets in comparison to EDSAS.

Different experiments are carried out to compare the performance results of EDSAS with the e-Sense algorithm. The e-sense algorithm is implemented in Matlab and run across the different Cyprus and India datasets. Since, e-Sense requires model construction through an offline training phase, a range of different training datasets are used in order to understand the effects on e-Sense's performance. Therefore, the rest of the section is structured as follows: Section 5.4.2.1 gives a detailed description of the e-Sense algorithm. Section 5.4.2.2 presents a comparison between EDSAS and e-Sense for self-trained datasets. Section 5.4.2.3 presents a comparison between EDSAS and e-Sense for cross-trained datasets.

5.4.2.1 Details about e-Sense algorithm

e-Sense is a stochastic scheduling algorithm that combines probabilistic data stream predictions with application data quality requirements to produce a sampling

schedule for each sensor node. A new data sensing approach is proposed by the authors in [57],[58], where the data measurements are made only when a state change is likely. With this approach, it is possible to miss certain state changes if data was not sampled at those time instances. Such missed state changes are referred to as *false negatives or misses.* Similarly, it is possible for this sensing approach to make a measurement when there is no actual state change. Such redundant sensing events are referred to as *false positives or hits.* Two quantities miss ratio μ and false hit ratio ρ are used to quantify the degradation in data quality and wasteful sampling respectively as follows[58]:

$$\mu = \frac{n_f}{n}, \rho = \frac{n_p}{n} \tag{5-13}$$

where, n_f and n_p denote the number of misses and false hits respectively, and n denotes the total number of sampling points.

The application's data quality requirement is expressed as a *tolerance level* $F_n \in [0,1]$ such that the miss ratio, $\mu \leq F_n$. It calculates the probability of a state change happening k (also denoted as $maxDelay$) time steps into the future using a data quality model [58] and based on the calculated state change probability, it determines a sampling probability for that instant. The data model is defined as follows [58]:

$$X_{i+k} = X_i + D_k \tag{5-14}$$

where, X_i denotes the data value at time instance i, X_{i+k} denotes the predicted data value at time $i + k$, i.e. k time steps forward from step i, and D_k represents the distribution of the difference between these values. Then, the model is completely defined once the distributions $D_m, m = 1, \ldots, k$ are known. One way to derive these distributions is by using a training data stream of length n: $X_i, i = 1, \ldots, n$ and for each value $m = 1, \ldots, k$ computing the histogram of the m-step value differences [58]:

$$D_m(i) = X_{i+m} - X_i, i = 1, \ldots, n - m \tag{5-15}$$

These distributions can be approximated by normal distributions, each of which needs only two parameters: the mean value, μ and the standard deviation, σ. Equation (5-14) can be rewritten as [58]:

$$\begin{aligned} X_{i+k} &= X_i + N(\mu_k, \sigma_k) \\ &= N(X_i + \mu_k, \sigma_k) \end{aligned} \tag{5-16}$$

The above model captures the data variation within k time steps as a random walk process. The statistical model described need to be constructed before it is stored on individual sensor nodes. Moreover, in order to capture changes in the dynamics of the observed data stream, the model may also have to be updated from time to time. The authors propose that the model construction and update should be carried out at base stations with sufficient computational and bandwidth resources.

The application uses the miss ratio bound F_n to limit the data quality degradation. Analogously, the energy wastage is bounded by using a false hit ratio limit, F_p. The system uses the false hit ratio limit, F_p to fine-tune the sampling probability. Whenever the observed false positive rate (false hit ratio ρ) becomes low, the value of F_p is incremented to allow more stochastic sampling events. On the other hand, when ρ increases to high values, the value of F_p is decremented to reduce the sampling probability and save energy.

5.4.2.2 Performance comparison for self-trained datasets

As a first step, the nodes are self-trained with an hour of data for e-Sense and results are compared against the performance of EDSAS. The various e-Sense parameters [57] are set as follows: the tolerance level F_n is set to 0.1, the $maxDelay$ is set to 5 and the false hit ratio ρ is set to 0.8. These e-Sense parameters are chosen to give a comparable performance to EDSAS for the parameters chosen in previous Section 5.4.1. The tolerance level is set to 10% in accordance with the missed ratio obtained from EDSAS. The $maxDelay$ is set to the same value as S_{max} for EDSAS and is used for training the e-Sense model for $maxDelay$ time steps as explained in the previous Section 5.4.2.1. The various results obtained for SF and MR across different nodes for both Cyprus (40% to 75% sampled data reduction with 11% to 5% missed changes) and Indian datasets (12% to 61% sampled data reduction with 15% to 8% missed changes) are tabulated in Table 5-3(a) and (b).

Table 5-3 SF, MR and SP obtained using e-Sense for various nodes for (a) Cyprus (b) India trials

(a)

Node Id	Sampling Fraction	Miss Ratio	SP
2	0.25	0.08	8.89
3	0.31	0.08	8.58
4	0.23	0.06	12.87
5	0.53	0.10	4.94
7	0.57	0.10	4.28
8	0.30	0.06	12.26
9	0.39	0.07	8.83
11	0.32	0.10	6.80
12	0.31	0.06	12.10
13	0.23	0.05	14.66
14	0.33	0.10	6.55
15	0.59	0.09	4.37
16	0.26	0.09	8.49
18	0.31	0.08	9.13

(b)

Node Id	Sampling Fraction	Miss Ratio	SP
1115	0.86	0.10	1.40
1118	0.60	0.12	3.28
1119	0.77	0.10	2.34
1121	0.64	0.15	2.48
1122	0.57	0.14	3.09
1123	0.87	0.10	1.30
1125	0.56	0.13	3.34
1126	0.39	0.13	4.65
1127	0.84	0.11	1.45
1129	0.88	0.08	1.64
1131	0.88	0.08	1.54
1132	0.82	0.10	1.91
1135	0.57	0.13	3.37
1137	0.71	0.10	2.94

The corresponding sampling performance metric obtained for various nodes for the Cyprus and the Indian datasets using both EDSAS and e-Sense is compared in Figure 5-10 and Figure 5-11 respectively. It can be seen that EDSAS yields higher sampling performance for all the nodes in case of Cyprus datasets as shown in Figure 5-10. In case of the Indian datasets, the sampling performance of EDSAS is only slightly higher than that obtained from e-Sense for most of the nodes for the given choice of parameters as shown in Figure 5-11.

It can be noticed from these graphs that there are certain nodes (5, 7, and 15) in case of the Cyprus datasets in Figure 5-10 for which the sampling performance value are lower than the rest of the nodes. This is because these are the nodes that exhibit more data variations and hence more data points need to be sampled in order to yield the data accuracy requirements. Similarly in case of the Indian datasets in Figure 5-11, for the nodes (1115, 1123, 1127, 1129, and 1131) that exhibit more variable data, sampling performance for EDSAS and e-Sense is lower than rest of the nodes.

It is important to estimate that the difference in sampling performance for the two algorithms is statistically significant or not in order to give conclusive results about the algorithm performance. A *paired sample t-test* [86] can be used to determine whether there is a significant difference between the sampling performances obtained from the two algorithms. The t-test gives a decision for the null hypothesis that the difference in sampling performance comes from a normal distribution with mean equal to zero and unknown variance. If the p-value is less than 0.05, it indicates that the null hypothesis can be rejected and the difference in sampling performance is statistically significant.

A paired sample t-test is carried out on the sampling performance results obtained from EDSAS and e-Sense for both the pollution datasets. For the Cyprus datasets with S_{max} set to 5 and δ set to 0.08 for EDSAS, p-value obtained for the results shown in Figure 5-10 is 0.0029 that indicates that the null hypothesis can be rejected. The difference in sampling performance is statistically significant and sampling performance of EDSAS is better than e-Sense for the Cyprus datasets.

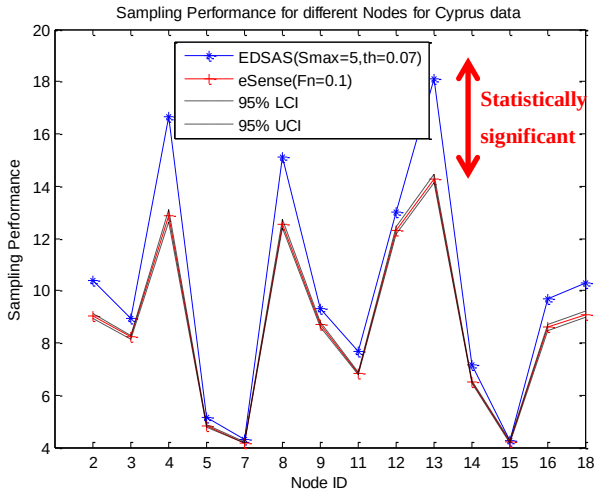


Figure 5-10 Comparison of SP obtained using EDSAS and e-Sense for Cyprus datasets

Similarly, the t-test is run for the Indian datasets for S_{max} set to 5 and δ set to 0.08 for EDSAS, the p-value obtained is 0.0765 that indicates that the difference in sampling performance is statistically not significant. This implies that performance of EDSAS is comparable to that of e-Sense for this choice of parameters. Next, a smaller error tolerance is tried for EDSAS for the Indian datasets because using smaller values of error tolerance yield higher sampling performance. The results are

shown in Figure 5-11 and the t-test is repeated for S_{max} set to 5 and δ set to 0.06 for EDSAS. The p-value obtained is 0.0006 that indicates that the difference in sampling performance is statistically significant for this choice of parameters.

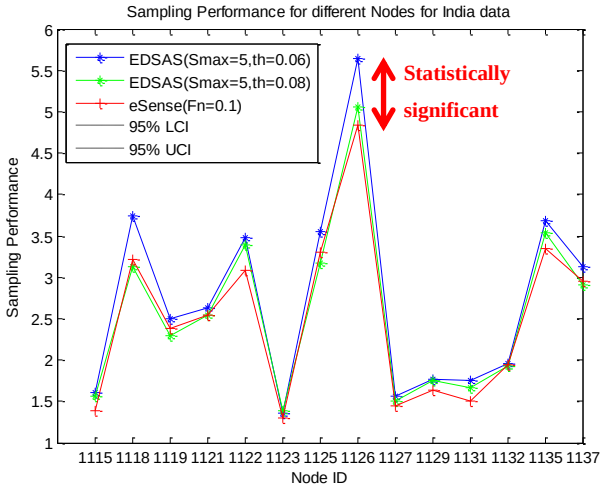


Figure 5-11 Comparison of SP obtained using EDSAS and e-Sense for India datasets

These experiments highlight that EDSAS outperforms e-Sense in terms of sampling performance for both the Cyprus and India datasets for a suitable choice of algorithm parameters. In-fact, the EDSAS is flexible enough and the sampling performance can be further improved by tuning the parameters (lowering the error tolerance or maximum step size) on the fly, whereas e-Sense lacks the flexibility and needs to be re-trained for model reconstruction and updates if the data changes significantly.

Next, in order to evaluate the measurement accuracy of the data sampled by EDSAS and e-Sense, average per minute deviations from the true mean is evaluated for the choice of algorithm parameters that are used in the previous experiments. The mean deviations provide an insight into the data accuracy levels obtained from the sampling algorithm. For both the Cyprus and Indian datasets the results are shown in Figure 5-12 and Figure 5-13 respectively. It can be seen from Figure 5-12 that EDSAS gives lower deviations than e-Sense for the majority of the nodes from Cyprus trial. Only for nodes 5, 7 and 15, deviations are more than e-Sense and this is because these nodes have more dynamic data and EDSAS is not able to keep up with the changes for the given choice of parameters. It is possible to get better measurement accuracy from these nodes by lowering the error tolerance. The mean

deviations for e-Sense are higher for most of the nodes and rise to as high as 3% for node 13.

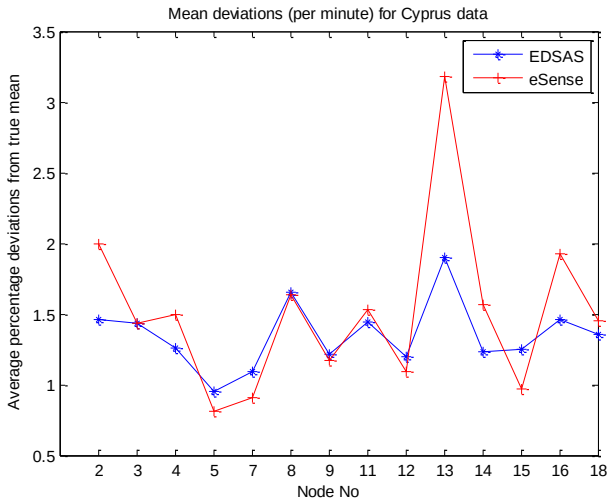


Figure 5-12 Comparison of average percentage deviations from true mean for Cyprus datasets

For the Indian datasets as shown in Figure 5-13, the mean deviations stay less than 1% for most of the nodes, with EDSAS giving lower mean deviations than e-Sense for across all the nodes.

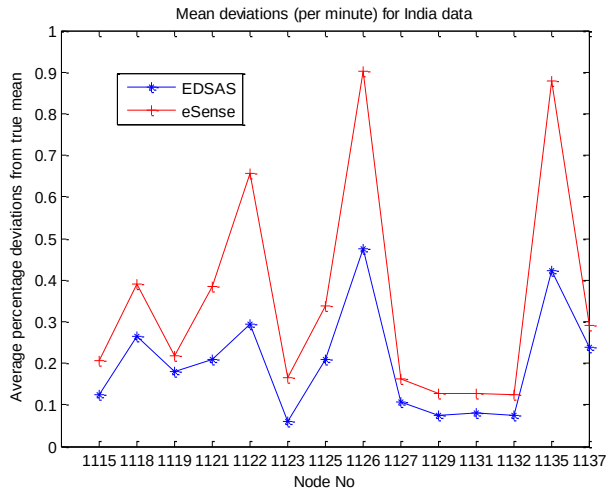


Figure 5-13 Comparison of average percentage deviations from true mean for India datasets

5.4.2.3 Performance comparison for cross-trained datasets

A good model is an essential pre-requisite in order to achieve reasonable performance from e-Sense. In case the model is not trained properly and is

inaccurate, the e-Sense performance is expected to deteriorate. Therefore, e-Sense is cross trained against data from different nodes and then the results are compared against EDSAS performance. These experiments are carried out to understand the impact of different data models on the performance of e-Sense. Table 5-4 summarises the information regarding different test and training datasets used for these experiments.

Table 5-4 Information about various test and training datasets

Test Datasets	**Node Id**	**Training Datasets**	**Node Id**
Dataset 1	7	Dataset 3	4
Dataset 2	1129	Dataset 4	7
		Dataset 5	16
		Dataset 6	1119
		Dataset 7	1122
		Dataset 8	1129

Dataset 1 and 2 are used as the test datasets for these experiments. Dataset 1 belongs to the Cyprus trial and dataset 2 belongs to the Indian trial. Datasets 3-8 are used as the training datasets for e-Sense. Datasets 3, 4, 5 belong to different nodes placed at different locations from the Cyprus trial while dataset 6, 7, 8 belong to different nodes placed at different locations from the Indian trial. The node locations for both the trials were shown in Figure 4-2 and Figure 4-5. For example, the test dataset 1 is trained against all the training datasets ranging from dataset 3 to dataset 8. This ensures that each test dataset is trained against nodes from both different locations and places.

Table 5-5 and Table 5-6 give the results for test datasets 1 and 2 respectively. For the test dataset 2, it can be seen from Table 5-6 that the highest sampling performance is achieved when dataset 2 is self-trained against dataset 8 (SP is shown in red), while for all the other training datasets, a lower value of sampling performance is obtained. This highlights the fact that higher sampling performance is achieved only in case the nodes are self-trained against their own data, while for all the other cases when the trained model is not adequate, the sampling performance obtained is lower. EDSAS sampling performance (1.64) for test dataset 2 still outperforms the self-trained e-Sense performance.

In the case of test dataset 1, it can be seen from Table 5-5 that a higher sampling performance is obtained when dataset 1 is cross trained against datasets 6 and 8 in addition to dataset 4 (self-trained). Sampling performance is shown in red for all the cases. With this test dataset 1, EDSAS sampling performance (4.28) still outperforms the self-trained e-Sense performance (4.13). Datasets 6 and 8 are able to give good performance because the training data is statistically similar to the test dataset 1, but the training datasets 3,4,5,7 do not give a better sampling performance.

Table 5-5 Results obtained from e-Sense for test dataset 1

Test Dataset	**Training Dataset**	**Sampling Fraction**	**Miss Ratio**	**Sampling Performance**
Dataset 1	Dataset 3	0.28	0.24	3.00
Dataset 1	Dataset 4	0.58	0.10	**4.13**
Dataset 1	Dataset 5	0.32	0.21	3.29
Dataset 1	Dataset 6	0.63	0.08	**4.42**
Dataset 1	Dataset 7	0.50	0.14	3.68
Dataset 1	Dataset 8	0.77	0.05	**4.80**
EDSAS performance for test dataset 1		0.57	0.10	4.28

Table 5-6 Results obtained from e-Sense for test dataset 2

Test Dataset	**Training Dataset**	**Sampling Fraction**	**Miss Ratio**	**Sampling Performance**
Dataset 2	Dataset 3	0.30	0.50	1.40
Dataset 2	Dataset 4	0.76	0.15	1.58
Dataset 2	Dataset 5	0.36	0.44	1.46
Dataset 2	Dataset 6	0.81	0.12	1.59
Dataset 2	Dataset 7	0.71	0.19	1.52
Dataset 2	Dataset 8	0.87	0.08	**1.61**
EDSAS performance for test dataset 2		0.88	0.08	1.64

It can be seen that choosing the correct cross training dataset can be very difficult since spatial variations across the different nodes can be very different leading to poor quality data model. Therefore, the model may need to be trained for individual nodes to achieve good accuracy levels. Moreover, the training of e-Sense model can only be done on a node with adequate memory resources and processing power. This means that a powerful node would require all the data from all the nodes in the

network, which is not a scalable solution for a wireless sensor network. The re-training of the models would also incur significant delays in the network to detect changes in the data.

5.4.3 Performance summary

The previous sections 5.4.1 and 5.4.2 provided an overview about EDSAS performance across different pollution datasets. EDSAS gives high sampled data reduction values of the order of 50%-70% for Cyprus datasets and 10%-50% across the Indian datasets. In both the cases, only around 10% of the changes are missed for the given choice of parameters. The parameters need to be set in accordance with the application requirements and they can be easily tuned on the fly to give a desirable performance.

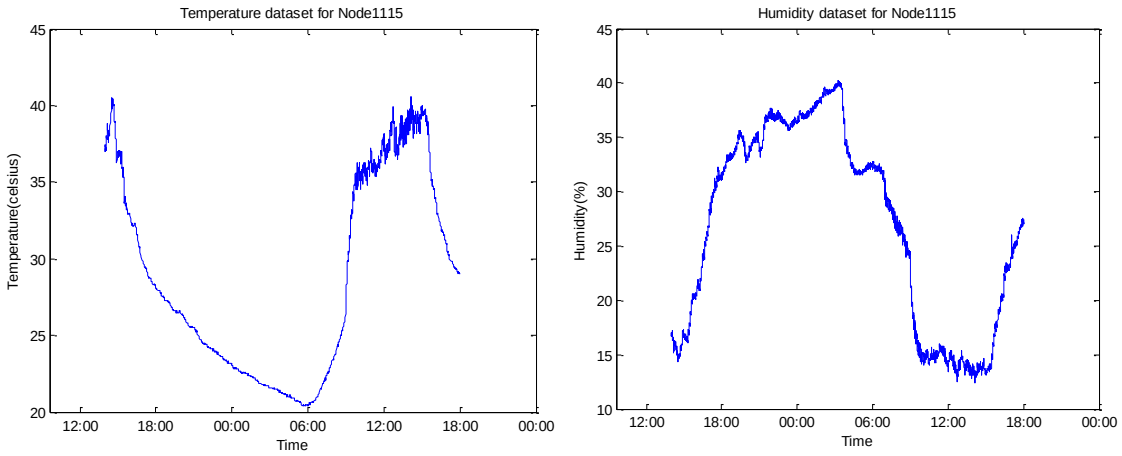
Further, it is seen that EDSAS gives higher sampling performance across datasets from both the locations and also gave lower mean deviations across the majority of the nodes in comparison to the model based sampling technique, e-Sense. EDSAS can achieve good performance without any training overhead and adapting automatically to the current level and trend existing in the time series using the recursive forecasting equations.

The main drawbacks for a model based approach are also highlighted by means of cross training e-Sense using different datasets from different locations. One of the pre-requisites for good performance in case of a model based approach is an accurate model and its regular maintenance. Model construction and updating in case of data changes for each individual node can constitute a major communication overhead in large wireless sensor networks.

5.5 Application of EDSAS to temperature and humidity datasets

EDSAS can be generalized to other datasets as well. In order to prove this, EDSAS is applied to the temperature and humidity datasets obtained from the Indian pollution trials. The sample temperature and humidity datasets from one of the nodes (node 1115) are shown in Figure 5-14(a)-(b). It can be observed from the datasets that they are less dynamic in comparison to the CO datasets and distinct locally

varying trends exist at different times of the day. Both EDSAS and e-Sense are applied to these datasets, changes for a relative threshold of 0.1℃ are evaluated for the temperature datasets and changes for a relative threshold of 0.1% are evaluated for the humidity datasets. The maximum step size ($S_{\max}$) varies from 10s to 1 min for EDSAS and correspondingly, the $maxDelay$ is set for e-Sense. The error tolerance (δ) is set to 0.01℃ for EDSAS. Tolerance level (F_n) is set to 0.1 for e-Sense. e-Sense is self-trained with an hour of temperature and humidity data from node 1115.



(a) (b)

Figure 5-14 Temperature and humidity dataset from a sample node

The results for SF and MR for the temperature dataset for both algorithms EDSAS and e-Sense are respectively shown in Figure 5-15 and Figure 5-16 and the corresponding SP comparison is shown in Figure 5-17. It can be seen that in case of e-Sense as shown in Figure 5-16, SF and MR become almost constant after a maximum delay of 20s. This indicates that for the given tolerance level, F_n, e-Sense cannot yield further improvement in the sampling performance. While in the case of EDSAS as shown in Figure 5-15, as the maximum step increases, there is constant performance improvement in the sampled data reduction values at the cost of data accuracy. It can be seen from Figure 5-17 that EDSAS sampling performance is higher than e-Sense for the temperature dataset at lower maximum step sizes. At higher step sizes from 50s to 60s, the sampling performance of EDSAS goes below e-Sense because the maximum step size becomes too large to capture the data dynamics for the given dataset.

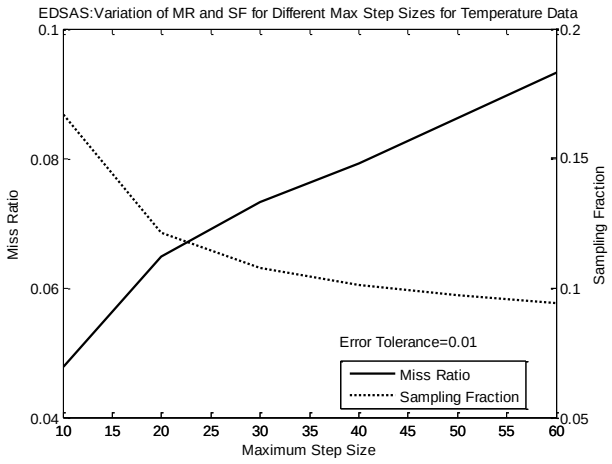


Figure 5-15 MR and SF for temperature dataset using EDSAS

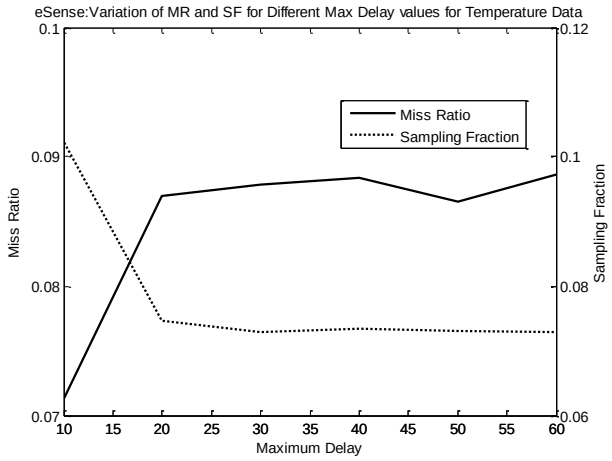


Figure 5-16 MR and SF for temperature dataset using e-Sense

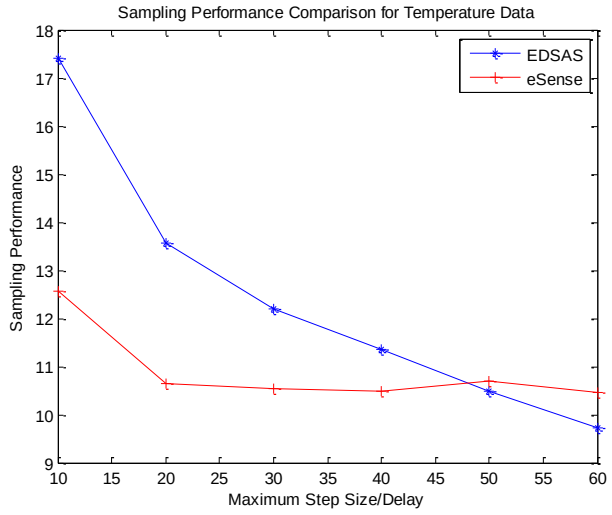


Figure 5-17 Sampling performance for EDSAS and e-Sense for temperature dataset

Similar experiments are carried out with the humidity datasets and results are shown in Figure 5-18 and Figure 5-19 for both EDSAS and e-Sense. The maximum step size varies from 2s to 10s for EDSAS and correspondingly, $maxDelay$ is set for e-Sense. The error tolerance for EDSAS is fixed to 0.01 and tolerance level for e-Sense is fixed as 0.1 as well. For the humidity dataset, EDSAS gives higher sampling performance than e-Sense at all the maximum step size values as shown in Figure 5-20.

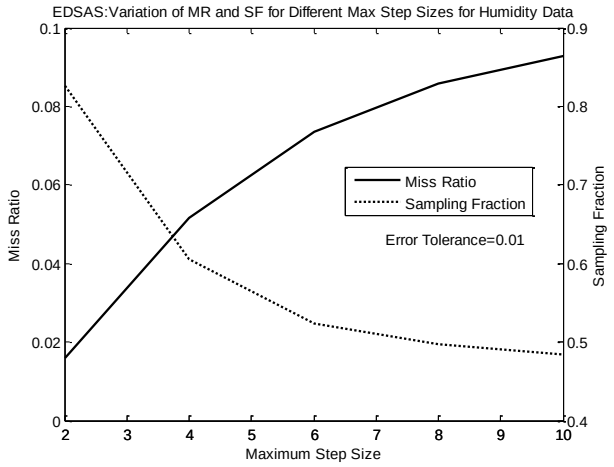


Figure 5-18 MR and SF for humidity dataset using EDSAS

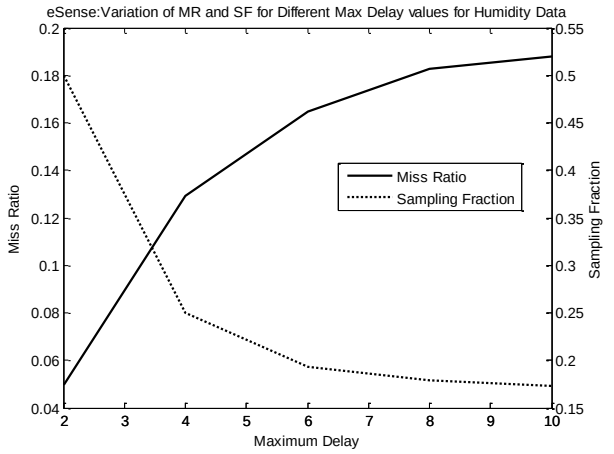


Figure 5-19 MR and SF for humidity dataset using e-Sense

It can be observed from Figure 5-19 that e-Sense is able to satisfy the given tolerance level only at very low maximum delay, i.e. 2s. For maximum delay values above 2s, the MR always stay above 0.1 and SF falls to as low as 0.2. This observation highlights the fact that if the e-Sense model is trained for inappropriate $maxDelay$ values that do not capture the data changes, the sampling

performance deteriorates. While in case of EDSAS, there is much tighter control on the sampling performance because of the use of two control parameters, and for different maximum step sizes too, MR is always less than 0.1 and SF varies from 0.8 to 0.5 as shown in Figure 5-18.

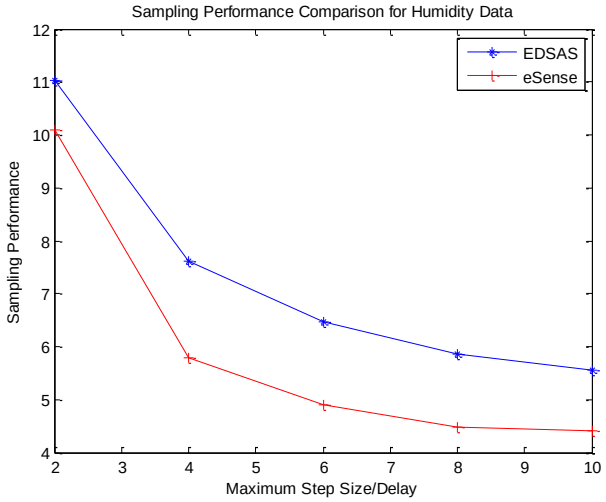


Figure 5-20 Sampling performance for EDSAS and e-Sense for humidity dataset

These experiments have further proven that EDSAS performs well across a range of datasets, not only for the pollution datasets and it gives adequate performance by tuning the algorithm parameters in accordance with the dataset. It does not rely on offline data training phase as in the case of e-Sense and can perform well without the requirement of an accurate underlying data model.

5.6 Chapter conclusions

In this chapter, a new design for temporal adaptive data sampling technique called *Exponential Double Smoothing based Adaptive Sampling* (EDSAS) has been proposed. It has been pointed out that while various adaptive sampling, model based and data reduction techniques in the temporal domain have been proposed so far in the research literature, there is a gap for light weight, real-time approaches based on time series forecasting. The chapter highlights the novel contribution to use the Wright's extension to exponential double smoothing (EDS) forecasting technique in the design of the proposed temporal adaptive sampling scheme. It explains that the time series prediction for irregularly sampled data can be used to sample the most appropriate data points in a time series. A change detection mechanism based on EWMA is also incorporated as a feedback mechanism in the proposed algorithm

design so as to avoid missing important changes and deterioration of measurement accuracy.

The performance metrics used for evaluating the sampled data reduction and measurement accuracy are explained and detailed results for the real pollution datasets from the Cyprus and the Indian trials are presented to test the hypothesis. The algorithm yields good results for sampled data reduction (50% to 70% for the Cyprus datasets and 10% to 50% for the Indian datasets), while not missing the important data changes (10%). The performance is also compared against a model based stochastic sampling technique called e-Sense and EDSAS is found to yield better sampling performance than e-Sense across both datasets for a suitable choice of algorithm parameters. The main advantages of EDSAS are its simplicity, flexibility, low overheads and ease of implementation (recursive design) and no requirement for offline training phase for model construction/update in comparison to the e-Sense algorithm. EDSAS has been shown to work across temperature and humidity datasets as well and gives good sampling performance in comparison to e-Sense.

Chapter 6
Application of EDSAS in spatial domain

6.1 Introduction

In the previous Chapter 5, EDSAS is proposed as a novel data sampling technique for adapting the sampling rate of a sensor node based on the existence of the temporal correlations in the sensed data. However, in a real world deployment with a large number of densely distributed wireless sensor nodes, the existence of spatial correlations between the nodes is another network characteristic that when exploited properly may lead to significant additional energy savings. Due to the presence of the spatial correlations amongst the readings of close sensor nodes, the measurements from a sensor node can be predicted from that of its nearby sensor nodes with high levels of confidence. While multiple specialised algorithms can be found in the research literature, which adapt the sampling rate either in the temporal domain or in the spatial domain, there is a lack of algorithms where temporally adaptive sampling techniques are extended into the spatial domain. In this chapter, therefore the EDSAS is further extended and applied to the spatial domain. The version of EDSAS used in the spatial domain is termed as *EDSAS-Spatial* (EDSAS-S) and the one used for temporal domain as described in previous Chapter 5 is termed as EDSAS-Temporal (EDSAS-T), and this notation is followed throughout this chapter.

To fully exploit the spatial correlations, the sensor nodes with similar observations can be grouped into a cluster. The sensor nodes within a cluster can be scheduled to sample alternatively to save energy and the reading of any sensor node can be approximated by other sensor node. In this work, two different clustering algorithms that use data correlations for node grouping – the Hierarchical Agglomerative Clustering based on Data Correlations (HAC-DC) [77] and the Affinity Propagation (AP) [78] are investigated and applied to the pollution datasets.

The outline of the remaining of Chapter 6 is as follows: different spatial sampling algorithms used in WSNs are studied and their main characteristics are presented in Section 6.2. Section 6.3 gives an introduction of the spatially interpolated datasets created from the real pollution datasets and their application for carrying out the spatial simulations. Sections 6.4.1 and 6.4.2 provide the details of the EDSAS-S algorithm and the various steps involved in its operation. Section 6.4.3 explains the two different spatial clustering algorithms used in the EDSAS-S design, the hierarchical agglomerative clustering based on data correlations and the affinity propagation clustering. The pros and cons of each of the techniques are investigated. Section 6.4.4 explains the data reconstruction for the spatially sampled data. Section 6.5 provides the detailed simulation results and performance comparison for the spatial clustering algorithms used for both the Cyprus and the India datasets. Section 6.6 explains the various metrics that are used to evaluate the performance of EDSAS-S and presents detailed performance analysis for different clustering algorithms and algorithm parameter values. Section 6.7 draws the conclusions of the chapter and highlights the benefits of EDSAS-S.

6.2 Adaptive sampling in the spatial domain

There are different approaches proposed for adaptive spatial sampling in WSNs in the research literature. In this section, a review of the major spatial adaptive sampling algorithms is provided in order to evaluate the design space and make well informed choices for a better spatial sampling algorithm design.

6.2.1 Spatial sampling techniques in WSNs

ASample [81] proposes a Voronoi based adaptive spatial sampling solution. This approach removes the unnecessary sensor nodes (SN) from regions of over-sampling and generates additional new sampling locations in the under-sampling regions to fulfil specified accuracy requirements. The first step towards an adaptive spatial sampling is for the SNs to locally check the fulfilment of the accuracy requirements. It is important to perform this in an efficient distributed manner as each redundant communication message reduces the battery lifetime of the involved SNs. This detection is based on the knowledge of SN's Voronoi diagram, which efficiently models each SN closest neighbourhood. Each SN evaluates whether any of its Voronoi neighbours sensor readings violates the required accuracy. In such case the

SN decides to place a virtual node (VN) at the Voronoi edge in order to separate the Voronoi neighbours whose values variance violates the required accuracy. The value of the VN is linearly interpolated and the Voronoi diagram is rebuilt including the new VN. As the approach is heuristic, the nodes are initially added as VNs, as some of them may be redundant. The redundant VNs can be removed so the SN proposes only the necessary sampling locations. The process iterates until the accuracy requirements are met by adding the VNs. For the situation of an evolving phenomenon, the SNs update their data to maintain accuracy. ASample employs a simulated model of a physical phenomenon and uses a mean square error based metric to quantify the measurement accuracy. This technique is mainly based on the approach of supplementing the network with additional resources, rather than optimizing the use of resources already present in the network, which is main focus in this thesis.

Another algorithm that follows a similar approach to spatial adaptive sampling is called Backcasting [82] and it operates by first having a small subset of the wireless sensors communicating their information to a fusion centre. This provides an initial estimate of the environment being sensed, and guides the allocation of additional network resources. Specifically, the fusion centre backcasts information based on the initial estimate to the network at large, selectively activating additional sensor nodes in order to achieve a target error level. The key idea is that the initial estimate can detect correlations in the environment, indicating that many sensors may not need to be activated by the fusion centre. Though their theoretical predictions for the balance of data accuracy and energy consumption are supported by means of simulated experiments, still they have not used any real datasets to evaluate it further.

The region sampling [83] approach begins by segmenting the network into several non-overlapping regions and performs sampling within each region. A *bottom-up partitioning* algorithm is used to partition the sensor network into regions. Each region computes the sample statistics, and sends the results back to the query node. The query node combines the partial aggregates from each region to approximate the query result. The sample statistics along with the sampling energy cost for each region are updated at the query node to compute the optimal sampling plan for each region. As the regions offer different sampling statistics and sensor sampling costs, different sample rates will be assigned to the regions in order to maximize the

accuracy, while constraining the query processing cost within a pre-defined budget. This algorithm considers only mean and median queries and does not employ temporal correlations. Also, the sampling schedule is created by a query node and is not decided in a real-time manner by the respective region heads.

Another distributed adaptive sampling technique for sensor based autonomic systems called SILENCE [84] is proposed to reduce redundancy in raw data through selective representation, and without compromising on the accuracy of the reconstruction of the phenomenon at the sink. *Similarity* and *correlation* in the sensed data is used on the fly to optimize the number of representatives reporting to the sink in a distributed manner, in both space and time domains. The proposed distributed solution enables each node to decide its state (or role) and sleeping schedule independently based on correlation and similarity of its own sampled data with that of the neighbouring nodes. This aggregated data helps a node determine whether to play the role of a *representative (*REP*)* and, consequently, to actively report data to the sink on behalf of a group of nodes; or to be an *associate (*ASSOC*)* to a REP and sleep. Putting nodes to sleep ensures that energy is not spent on packet receptions as well as sensing. The representation of a group of ASSOCs by a single REP is possible only due to the use of similarity along with correlation. If only correlation is considered, a REP would represent nodes that experienced only a correlated trend in variation of the manifestation compared to its own and not similar values. The REPs also exploit the temporal correlation characteristics of their sensed data to adapt the rate of control message broadcasts and data transmissions to the sink. Furthermore, ASSOCs wake up periodically and can identify or track any variation in the spatial distribution over time and change their state accordingly. This technique uses both temporal and spatial correlations to achieve adaptive sampling and it has shown to work with temperature and humidity sensors in the datacentres, which are both slow varying variables in comparison to the dynamic pollution datasets. This technique does not employ clustering mechanisms for finding representative nodes and the local message overheads for choosing REP and ASSOC nodes can be very large in case of fast changing data dynamics.

It can be observed that all of the above mentioned spatial sampling algorithms are specifically for spatial sampling and do not adapt the sampling rate in the temporal domain. Hence there is a need for an algorithm, which exploits both the temporal and

the spatial data characteristics, while taking care of the trade-off between energy savings and measurement accuracy. Also, all the spatial sampling techniques presented mainly try to optimize the sensor energy costs only. The communication energy costs for data collection at the sink are not accounted for by these spatial sampling techniques. In the literature, there are specific data collection frameworks that use temporal and spatial correlations simultaneously and try to optimize both the sensor and communication energy costs. The most prominent data collection frameworks that exploit both the temporal and spatial correlations are mentioned in the next section.

6.2.2 Data collection frameworks based on temporal and spatial correlations

One of the approaches that exploit spatial and temporal correlations is proposed in [85] and it is called Adaptive Sampling Approach to Data Collection (ASAP). The authors provide a distributed sensor-driven clustering scheme to construct a network organization that achieves the objectives of energy awareness and high-quality data collection. The scheme consists of two phases: a cluster head selection and a cluster formation performed after every clustering period denoted by τ_c. During the cluster head selection phase, the nodes decide if they should assume the cluster head role based on the cluster count factor and the energy level. Next, the cluster formation phase organizes all nodes in the network into clusters using two major steps: a message circulation and a cluster engagement. In the message circulation step, after nominating itself as a cluster head, the node sends out a message containing the value of its readings. The message is continuously relayed by neighbours within a given number of hops of the cluster head. In the cluster engagement step, upon receiving the advertisement from a cluster head, the sensor node calculates the attraction score of this cluster head and joins the cluster with the highest score. The metrics used to group nodes within the same cluster include the similarity of sensor readings and the hop count.

A predefined data collection tree is constructed on top of the network for the communication between the sensor node and the base node. The sink node is the root of the data collection tree. However, the data collection tree becomes a liability of the scheme, as keeping the tree for a long time could exhaust the energy of the nodes

belonging to the tree. Another drawback is the method of selecting the cluster head based on probability. The clustering process does not involve adaptability based on the correlation changes. The clusters should be dynamically able to self-reorganize when the spatial locality pattern changes without being a network burden. If a node receives a cluster formation message from only one cluster head, it has to join this cluster regardless of the dissimilarity between its sensor reading and that of the cluster head. This causes the variance of the readings between all the nodes in the same cluster to be relatively large; therefore, affecting the sampling accuracy.

Once the clustering process is performed, not all nodes in the same cluster are requested to sample the environment: the correlation-based sampler selection is performed at each cluster head and aims at determining those sampler nodes that best capture the spatial and temporal correlations among the other sensor readings. Moreover, probabilistic models for non-sampler nodes are built. Finally, ASAP collects sensor readings from only a subset of nodes (sampler nodes) that are previously selected. The values of non-sampler nodes are predicted using the probabilistic models built in the previous step; clusters are dynamically changed after each predefined schedule update period. ASAP is chosen as the reference sampling technique for comparison against the spatial sampling technique proposed in Chapter 7. Therefore, more details about the ASAP technique can be found in Section 7.4.2.1.

Liu et al. propose an Energy Efficient Data Collection (EEDC) framework to continuously collect data in sensor networks [87] by exploiting the spatiotemporal correlations. To exploit the spatial correlation, the sensor nodes with similar observations are partitioned into a cluster. Within each cluster, the reading of any sensor node can be approximated by any other sensor nodes within an error bound. Therefore, the sensor nodes within a cluster can be scheduled to work alternatively to save energy. To exploit temporal correlation, piecewise linear approximation technique is adopted, i.e. approximating the time series with a sequence of line segments. In order to minimize the energy consumption on data transmission with certain accuracy at the sink node, the problem is modelled as an optimization problem, called the Piecewise Linear Approximation with Minimum number of Line Segments (PLAMLiS) problem.

The basic assumption is that all the sensor nodes are within a single-hop radio transmission to the sink node, or to a local centre. The clustering scheme proposed in

EEDC employs two metrics, the magnitude *m-dissimilarity* and the trend *t-dissimilarity*, to evaluate the differences in the time series of the sensed data from two sensor nodes. The EEDC clustering scheme uses a centralized algorithm to partition the sensor nodes into exclusive clusters such that, within each cluster, the pair-wise dissimilarity measures of the sensor nodes are below a given threshold. Based on the sensed values received from all the sensor nodes in the network, the sink calculates the pair-wise dissimilarities between each pair of sensor nodes, and then runs the clustering algorithm to partition the network. The overall problem is solved by modelling it as a clique-covering problem in graph theory and uses a greedy algorithm to obtain the result. The primary limitation of the scheme is the assumption of the single-hop network architecture. This assumption is impractical and hard to justify in large scale wireless sensor networks. Another drawback is that the clustering algorithm is centralized and runs at the sink. This means that data from all sensor nodes have to be sent to the sink. The sink stores and processes a huge amount of data to partition the network. As it can be observed from the above mentioned techniques, the clustering mechanism for spatial sampling should be decentralized and adaptive to the spatial correlation changes so as to enhance the practicality.

Based on the study above, it can be concluded that a combined temporal and spatial adaptive sampling technique is required. The technique should incorporate a distributed spatial clustering mechanism while adequately capturing the temporal characteristics. This requirement has motivated the extension of the proposed temporal sampling algorithm, EDSAS in the spatial domain using a distributed hierarchical agglomerative clustering.

6.3 Spatial interpolation

In order to evaluate any spatial algorithm, spatial data is required and the data obtained in this work through the pollution trials is limited by the total number of nodes (fourteen) available and deployed. Hence the pollution data obtained from sensor nodes needed to be spatially interpolated to generate a number of intermediate spatial points (nodes) and their corresponding pollution levels. In general, interpolation refers to the process of estimating the unknown data values for specific locations using the known data values for other points.

In the current work, *TriScatteredInterp* [67] from MATLAB is used to perform spatial interpolation in the two dimensional space. A scattered data set defined by locations X and corresponding values V can be interpolated using a Delaunay triangulation of X. The *Delaunay triangulation* [68] of a set of points is a triangulation such that the unique circle circumscribed about each triangle contains no other points in the set as shown in Figure 6-1.

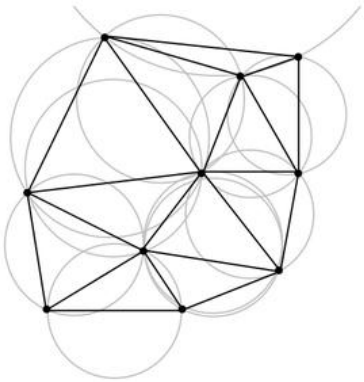
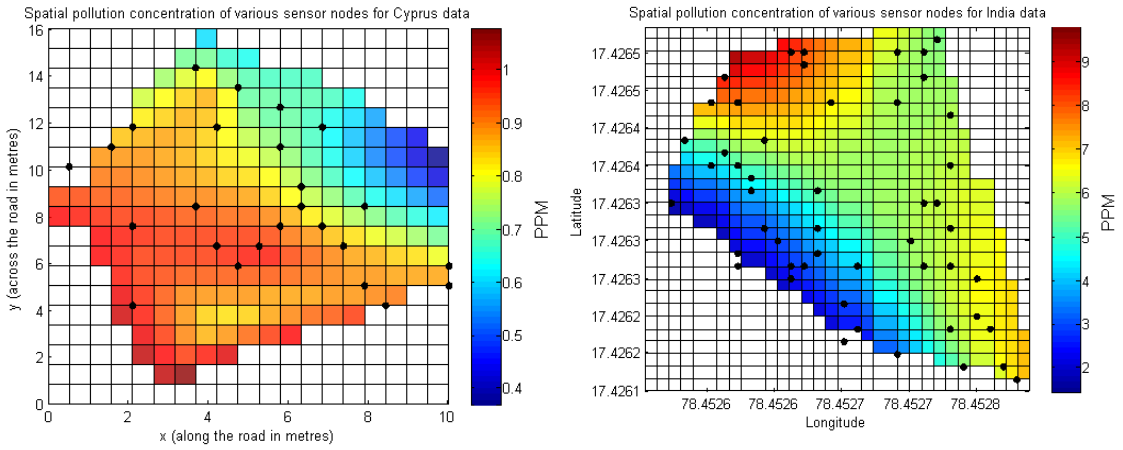


Figure 6-1 Delaunay triangulation with circumcircles

This produces a surface of the form $V = F(X)$. The surface can be evaluated at any query location QX consisting of x and y location coordinates, using $QV = F(QX)$, where QX lies within the convex hull of X. The *convex hull* of a set of points is the smallest convex set containing all points of the original set.

The spatial interpolation for the original set of nodes used in the pollution trials as mentioned in Section 4.2 is carried out using *TriScatteredInterp*. Once the spatial interpolation is done, time series data is generated for a number of other unknown node locations in the interpolated spatial field. Different network topologies can be generated from these interpolated spatial locations by choosing random node locations and their corresponding pollution data. These different network topologies are further used to carry out the simulations for performance measurements of both the spatial clustering and sampling algorithms.

Figure 6-2(a) and (b) show the interpolated spatial field created at a particular time instant using the known node locations and their data for both the Cyprus and India datasets. Random node positions (shown by the black dots) can be selected within this spatial field to define different network topologies. The colour bar indicates the intensity of the pollution levels being measured by different sensor nodes.



(a) (b)

Figure 6-2 Spatially interpolated (a) Cyprus (b) India data

It can be easily visualised from these figures that the nodes located close to each other measure similar pollution levels and can be clustered together. A single representative node or a fraction of nodes in the each cluster can be selected to sample the data and the data for the remaining nodes can be estimated using the data from the representative nodes. This principle is applied for EDSAS-S and its detailed description is given in the following Section 6.4.

6.4 EDSAS-S for spatial sampling

Due to the presence of spatial correlations in the data sensed from the adjacent nodes, if the sensor nodes are clustered based upon data correlations and either a single representative cluster node or a fraction of the cluster nodes report the data, they cannot only provide an overview of the sensory data distribution in the adjacent area, but they can also contribute in terms of sensor and communication energy savings on behalf of the non-sampling nodes in each cluster. Therefore, a spatial clustering based on the data correlations is included as a pre step to the spatial sampling algorithm proposed in this work. Two different spatial clustering approaches, the distributed hierarchical agglomerative clustering and the affinity propagation clustering, are investigated in this work. The hierarchical agglomerative clustering algorithm uses neighbourhood communication to create the clusters in a distributed manner, while the affinity propagation algorithm focusses on finding the best representative nodes and their corresponding cluster members based upon a

correlation distance similarity matrix in a centralized manner. Though keeping in mind the limited computing resources on each sensor node and the multi-hop communication scheme of sensor networks, it might be better to perform the clustering in a distributed manner without incurring much overhead. The section 6.4.1 provides an overview of the EDSAS-S algorithm design. The network architecture and detailed algorithm description are presented in Section 6.4.2. The description of the spatial clustering algorithms is presented in Section 6.4.3. The reconstruction of the time series for non-sampled nodes is described in Section 6.4.4.

6.4.1 Brief overview of EDSAS-S

The spatial algorithm is designed to run in consecutive time cycles of full and adaptive sampling. The time cycle duration is a network wide parameter and is termed as time scale, s. Time cycles can be chosen depending on the application requirement, but for the pollution monitoring application due to the fast changing dynamics, the time cycles are chosen in the range of 10 min to half hour. During the full sampling cycle, all the nodes sample data using the temporal adaptive sampling algorithm EDSAS-T as explained in Chapter 5. At the end of the full sampling cycle, spatial clustering is initiated and based on the correlations in the data gathered in the full sampling cycle; clusters are formed. Once the clustering is performed, the adaptive sampling cycle is initiated. Within each of the newly formed clusters, a single representative node or a fraction of nodes within the cluster are used to run EDSAS-T and gather the data, whereas the remaining nodes are put to sleep. The data from the non-sampling nodes can be estimated offline using data from the full sampling nodes.

The main premise in this sampling algorithm is that both the sensor energy savings and the communication savings accrued by putting a fraction of nodes to sleep within different clusters exceeds the clustering message overhead incurred at the end of each full sampling cycle. Figure 6-3 illustrates the EDSAS-S sampling operation:

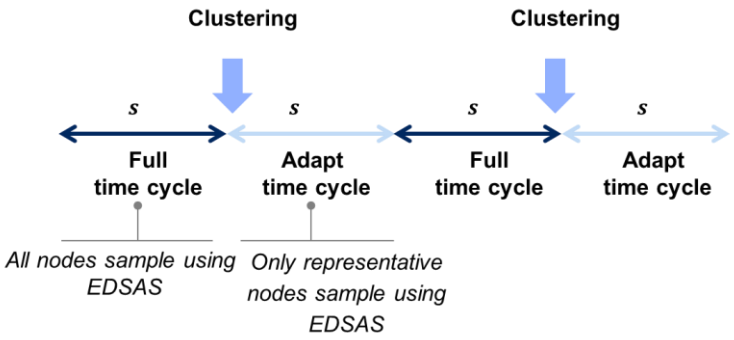


Figure 6-3 EDSAS-S algorithm operation

6.4.2 Network architecture and detailed EDSAS-S algorithm

The EDSAS-S algorithm is assumed to run in a network architecture comprising of a wireless network formed by N sensor nodes. Each node in the network is denoted by p_i, where $i \in \{1, \ldots, N\}$. Each node is assumed to be able to communicate only with its neighbours, that is, the nodes within its communication range, R. The set of neighbouring nodes of node p_i is denoted by $nbr(p_i)$. In each full sampling cycle, all the nodes, N, sample data using EDSAS-T for the whole duration of the full cycle. Based on the temporal correlations, EDSAS-T yields several sampled data points for the entire time scale for each node p_i. In order to carry out the clustering process, exchanging whole sampled data series amongst nodes can be quite expensive. Hence, the time scale is divided into a number of intervals using a network wide parameter, correlation interval, CI. Average values are computed across all the CI intervals and at the end of the full sampling cycle, each vector containing the averaged data readings, $D[p_i]$ is exchanged amongst the neighbouring nodes. These vectors are used for correlation computation for carrying out the clustering process. The clustering process will yield clusters, C_j, where $j \in \{1, \ldots, M\}, M < N$ in which the nodes are grouped based on similarity of data correlations. In case of hierarchical agglomerative clustering, a correlation threshold, C_{th}, is used as the threshold value for creating the clusters, i.e. only for the nodes whose data correlation values are greater than C_{th} are clustered together. In case of the affinity propagation clustering, the algorithm is capable of finding the best representative nodes based on the similarity matrix containing correlation distances between different nodes. More details on the clustering algorithms are given in Section 6.4.3.

In each adaptive sampling cycle, for each of the clusters, C_j obtained from the clustering process, a fraction of the cluster nodes are chosen as sampler nodes (S_j) based on the remaining energy levels and rest of the nodes become non sampler nodes (NS_j). The selection of the sampler nodes (on the basis of the remaining energy levels) will ensure that all the nodes get a fair distribution of sampling and the same set of nodes need not to be involved in sampling at each adaptive cycle. A system wide parameter, the sampling node fraction, $\sigma \in (0,1]$, is introduced to define the average fraction of nodes that should be used as samplers. It is ensured that there is at least one sampler node for each cluster. Another approach that can be applied during the adaptive cycle is that only a single representative node samples the data points while the remaining nodes are put to sleep in the cluster. In the case of affinity propagation clustering, which yields a single best representative node for each cluster, this approach is applied.

The sampler nodes keep sensing the data using EDSAS-T for the whole duration of the adaptive sampling cycle, while the non-sampler are put to sleep for the entire duration of the adaptive sampling cycle. During the adaptive cycle, data from only the sampler nodes need to be sent to the base station. The data for the non-sampler nodes can be estimated using the polynomial regression relationship derived between the nodes from the data collected at the previous full sampling cycle. More details on data reconstruction are in Section 6.4.4. At the end of the adaptive cycle, all the nodes revert back to the full sampling time cycle and this process is carried on as long as sampling needs to take place.

6.4.3 Spatial node clustering based on data correlations

The choice of an appropriate clustering algorithm is an important factor for an adequate design and performance of the spatial sampling algorithm. Therefore, in this section, a brief introduction to clustering and a survey of the most relevant clustering techniques used in WSNs is provided. This is followed by the details of the two clustering algorithms that are chosen in this thesis and both the algorithms follow very different approaches to cluster the nodes based on their data correlations.

6.4.3.1 Background on clustering

Clustering is a well-established technique for reducing energy costs in WSNs [69]. In this technique, the sensor nodes are grouped into disjoint sets, with each set managed by a designated cluster-head (CH), selected from among the sensor nodes. Most of the clustering protocols in WSNs are top-down approaches, which first formulate a global knowledge of a WSN. Based on the global knowledge of the network and predefined methods, the protocols first build the upper level of clusters by selecting certain nodes as CHs. Then they group the rest of the nodes into the designated cluster as cluster members. Many algorithms either randomly select CHs or choose well selected CHs.

Low-Energy Adaptive Clustering Hierarchy (LEACH) [70] is a classic probabilistic clustering algorithm. It aims to distribute the traffic load evenly among the sensor nodes and reduce the network energy consumption. The LEACH algorithm proceeds in rounds. In each round, each sensor node independently decides whether or not to become a CH according to a probability function. On average, this function allows each node to become a CH for a similar period of time assuring fair balancing of energy consumption among all the nodes. Although LEACH performs local data fusion to compress the cluster information, it does not consider data correlations when forming optimal-sized clusters.

There are other distributed clustering methods in sensor networks, such as ELink [71] and Distributed Single pass Incremental Clustering (DSIC) [72]. Elink uses an autoregressive model for the time series obtained at individual nodes, and then, based on a communication graph, a local clustering starts from a set of nominated root nodes and expands to include other nodes if the Euclidean distances between their model coefficients are less than a pre-defined threshold. However, the performance of Elink is limited because each cluster is coarsely represented by the feature of the cluster root rather than the data characteristics of the whole cluster. DSIC technique uses a hierarchical structure of sensor networks as the underlying infrastructure, where the sensor nodes are self-organized into physical clusters with one node selected as a cluster head for each physical cluster, and the cluster heads form a routing tree back to the gateway. DSIC technique works in two phases. In the first phase, the time series produced at sensor nodes are first transformed to a compact representation using the Haar wavelet transform [73] and the selected wavelet

coefficients are sent to a cluster head. Upon receiving the new data from its children, a cluster head first reconstructs the time series and then incrementally construct a local clustering model based on the Dynamic Time Warping (DTW) [72] distances. In the second phase, the data clusters are merged across different physical clusters along the routing tree until the gateway obtains a global clustering model. In both of these algorithms, the clustering processes are dependent on some communication infrastructure, such as a quad-tree or routing tree. When the data distribution is inconsistent with the communication topology, the clustering quality may be heavily influenced. Further, in [74] the authors show that the cluster-based sensor networks generally outperform non-clustered WSNs, and clusters should comprise nodes with highly correlated data-readings. Therefore, a clustering algorithm that does not depend upon a fixed communication topology and can adapt the clusters according to the data distribution with minimum overheads is required. The hierarchical agglomerative clustering based on data correlations is suitable for this purpose. Its clustering performance needs to be benchmarked against a good performance clustering algorithm and the affinity propagation is chosen for this purpose. These algorithms follow very different approaches to cluster the spatially co-located nodes based on the correlations in their data measurements. Both of the clustering techniques are described in more details in the following sections 6.4.3.2 and 6.4.3.3:

1. The hierarchical agglomerative clustering based on data correlations (HAC-DC) [75],[77] is a bottom-up distributed clustering approach. It uses the statistical features of the data from the nodes for cluster formation and can capture the data distribution well during the distributed and hierarchical merging process.

2. The affinity propagation (AP) clustering [78] is a recently proposed clustering algorithm that performs better than the classical K-means clustering [80]. This approach uses real valued messages to find out the best set of cluster heads in an iterative manner. This technique gives the optimal set of clusters based upon data correlations and serve as a benchmark for evaluating the clustering performance.

The clustering behaviour of the two clustering algorithms is studied in details and compared against each other in Section 6.5 and their impact on the sampling performance is studied in Section 6.6.1.

6.4.3.2 The hierarchical agglomerative clustering based on data correlations (HAC-DC)

The hierarchical agglomerative clustering (HAC) [77] is a conceptually and mathematically simple clustering approach to data analysis. With the bottom-up HAC approach [76],[77] , sensor nodes collaborate and build clusters based on one-hop neighbourhood information. Each sensor node is treated as an initial cluster and then geographically adjacent one-hop clusters gradually are merged based on the similarity in their readings. This bottom-up process is efficient because spatial correlations exist in the real world sensory data, and therefore, the computation and communication should be limited mostly amongst the proximate clusters. As a result of this the clusters will not grow too large and clustering process will take less amount of time. Thus the bottom-up approach can be a better way to implement self-organization, scalability and flexibility.

In this work, *correlation coefficients* are computed to find the similarity between data from two nodes. Correlation provides a measure of the relationship between measured data values. The correlation coefficient $\rho_{X,Y}$ between two random variables X and Y with expected values μ_X and μ_Y and standard deviations σ_X and σ_Y is defined as follows[32]:

$$\rho_{X,Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y} \qquad (6\text{-}1)$$

The correlation values, $\rho_{X,Y}$ are always in the range [-1, 1], -1 and 1 representing strongest negative and positive correlations. A value of 0 implies that the two time series are not correlated. In this thesis, the absolute values of correlation coefficients are used and hence, $\rho_{X,Y}$ lies in the range [0,1]. The hierarchical agglomerative clustering used in this work is termed as *Hierarchical Agglomerative Clustering based on Data Correlations* (HAC-DC) because of the use of data correlations.

The main idea behind HAC-DC [75],[76],[77] is that a node only needs one-hop neighbour knowledge to build the clusters. HAC-DC adopts the ‘‘general assumptions” of WSNs as follows:

1. The nodes in the network are quasi-stationary.
2. The nodes are left unattended after deployment.

3. Each node only has local information or the identification of its one-hop neighbour nodes.
4. All nodes have similar capabilities, processing, communication and initial energy.
5. The propagation channel is symmetric.

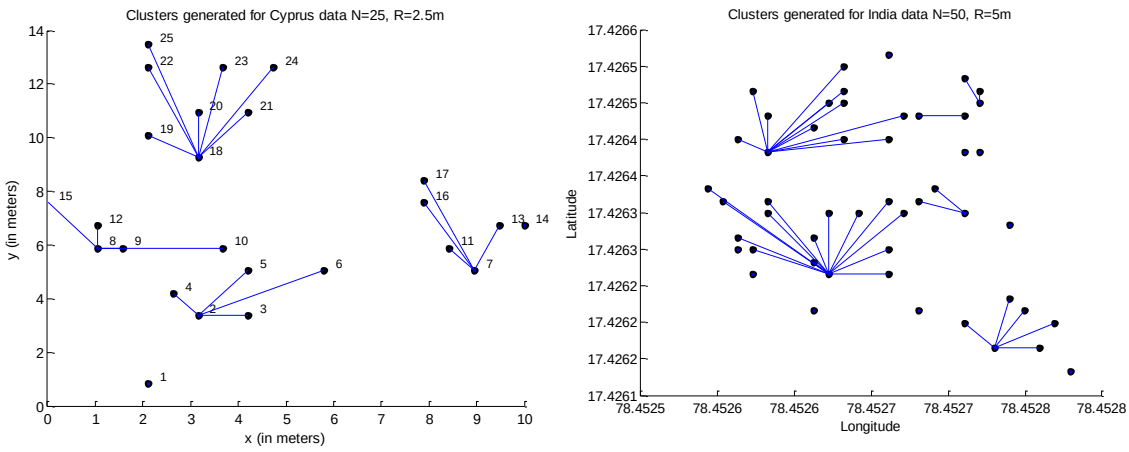
The clustering algorithm runs in multiple rounds. Initially, each node treats itself as an active cluster. Then, similar adjacent clusters are merged into larger clusters round by round. In each round, each cluster will try to combine with its most similar adjacent cluster simultaneously. Two clusters can be merged only if both consider each other as the most similar neighbour. In this clustering mechanism, emphasis is on generating the network partitioning/node grouping that reflects the data distribution well and there are no designated CHs. Any cluster node can be used for the data sampling task.

At the beginning of the first round each node, p_i has a state set to $ACTIVE$ and each node is assumed to form its own cluster C_i, with itself as a member. At the same time instant, all nodes broadcast a $HELLO$ packet with a vector of averaged data points, $D[p_i]$. All the nodes, p_j, that receive the $HELLO$ packet will compute the correlation between their vector of averaged data points and the received vector. If the correlation value is within the correlation threshold, C_{th} and the node-id of p_j is greater than that of p_i, the receiving node sends an $ACCEPT$ packet to the sender and sets its state to $PASSIVE$. In this manner, the clusters formed will be centred on the nodes with the lowest node-ids. Once the previous sender receives the $ACCEPT$ packet, logically the clusters C_i and C_j merge together to form a new cluster with the node p_i as the cluster head. The new neighbour list of node p_i is updated to contain the one hop neighbours of node p_j, that are not already present in the neighbour list of node p_i. At the end of round one, all the one–hop merging between individual nodes will be complete and next round of merging is initiated.

At this round, the still $ACTIVE$ clusters look for merging with the one-hop clusters formed at the previous round. So, they again send $HELLO$ packets to their still $ACTIVE$ neighbours. The neighbours receive the $HELLO$ packet and compute the correlations and depending on whether the correlation criterion is satisfied or not, these clusters merge after sending an $ACCEPT$ packet to the previous sender and

turning their state to $PASSIVE$. If no new merges take place in this round, then the clustering process terminates, otherwise, the merging in successive rounds continues as long as there are still $ACTIVE$ nodes remaining.

Thus, it can be seen HAC-DC provides a simple and efficient mechanism for real-time data analysis for a large amount of data. It can provide an idea about the data distribution around the network by partitioning the network into contiguous regions with similar readings without relying on any communication infrastructure. Figure 6-4(a) and (b) show clusters produced upon running the HAC-DC on the spatially interpolated datasets. The number of nodes is set to 25 and the transmission radius to 2.5m for the Cyprus datasets while the number of nodes is set to 50 and the transmission radius to 5m for the India datasets. The reason for choosing these network settings is explained further in Section 6.5.



(a) (b)

Figure 6-4 Clusters generated using HAC-DC for (a) Cyprus (b) India data

The number of clusters generated for the Cyprus dataset is 6, while the number of clusters for the India dataset is 15 using HAC-DC. It can be seen that there are a number of single node clusters since either these nodes are out of the transmission range of other nodes or their data correlations are significantly different from each other. This behaviour of HAC-DC is evident because of the distributed execution. Also it can be noted from Figure 6-4(a) and (b) that the clusters are centred on the lowest node id. For example, in Figure 6-4(a), the clusters are centred on nodes 2, 7,8,18.

6.4.3.3 The Affinity propagation (AP) clustering

The Affinity propagation [78] is a recent clustering method that is shown to produce clusters in much less time, and with much less error than the traditional clustering techniques, such as K-means clustering [17],[80]. The K-means clustering method aims to partition a set of data into K clusters in which each data point belongs to the cluster with the nearest mean value. Although, the main idea of the K-means clustering is quite simple, it needs to determine the number of clusters (the value of K) in advance, and it consumes too much processing time until the best results are selected. Due to those harsh requirements, it is not a feasible method for a large wireless sensor environment. The AP clustering can solve the above issues.

The AP clustering can be utilized to identify a relatively small number of cluster centres (i.e. *exemplars*) to represent all the points in a data set. Exemplars are the data points that are chosen to be the cluster centres. They are representative of themselves and some other data points that belong to the same clusters with them. In AP clustering, each data point can be viewed as a node in a network and simultaneously considered as a potential exemplar at first, and then real-valued messages are recursively transmitted along the edges of the network until a good set of exemplars and corresponding clusters emerges. At any point in time, the magnitude of each message reflects the current affinity that one data point has for choosing another data point as its exemplar and therefore this method is called *Affinity Propagation*.

In this thesis, the goal is to cluster the nodes on the basis of similarity in the pollution data measurements and identify the most representative nodes in the network that can be sampled at higher rates. Therefore, AP clustering seems to be a good candidate to provide an idea about the optimal clustering performance. A review of the mathematical model of the AP [78]clustering approach is given below.

There are two input arguments for the AP clustering algorithm: *similarity* and *preference*. Affinity propagation takes an input function of similarities, $s(i,k)$, where $s(i,k)$ indicates how well suited a data point k is to be the exemplar (i.e. cluster centre) of a data point i. If the data are real-valued, a common choice of similarity function could be the negative Euclidean distance between data points, so that a maximum similarity corresponds to the closest data points. AP clustering can be applied to use general notion of similarity, and the similarities can be positive or

negative depending on the way in which the definition of similarity is appropriate for the application. In this thesis, similarities are computed based on the combination of Euclidean distance and correlation distance. The Euclidean distance (ED) is computed for a pair of sensor nodes, i, k by taking their two-dimensional x and y location coordinates into consideration. The ED is included so that only the geographically proximate nodes with similar correlations get clustered together. The c*orrelation distance* (CD) between the data from a pair of sensor nodes, i, k is given by the following relationship:

$$CD = 1 - |\rho_{i,k}| \tag{6-2}$$

The value of correlation distance varies from [0, 1] where a value of 0 corresponds to maximum correlation ($|\rho_{i,k}|$ is 1) and a value of 1 denotes no correlation ($|\rho_{X,Y}|$ is 0) between a given pair of nodes. So, the similarities $s(i, k)$ can be defined as the sum of the ED and CD as follows:

$$s(i, k) = ED + CD \tag{6-3}$$

Each data point i has a self-similarity, $s(i, i)$, which reflects the prior suitability of data point i to be an exemplar and influences the number of exemplars that are identified. The self-similarity is also called “preference" and is the second input parameter for this algorithm. Assigning a data point to a larger or smaller preference (self-similarity) value will either increase or decrease the possibility of the data point becoming an exemplar. In the beginning, AP clustering considers all data points as potential exemplars. So if one wants to make sure all data points are equally suitable as exemplars and there is no inclination toward particular ones as exemplars, the preferences of all data points should be set to the same value. In addition, the preference value can control the number of clusters that are generated.

In the AP clustering method, two kinds of messages are exchanged between the data points: *responsibility* messages and *availability* messages. The responsibility message, $r_{res}(i, k)$, is sent from the data point i to the candidate exemplar data point k. A non-exemplar data point i informs each candidate exemplar whether it is suitable for joining as a cluster member or no, as shown in Figure 6-5(a). The message, $r_{res}(i, k)$ indicates how well suited the data point k is to be the exemplar for data point i, taking into account the other competing potential exemplars.

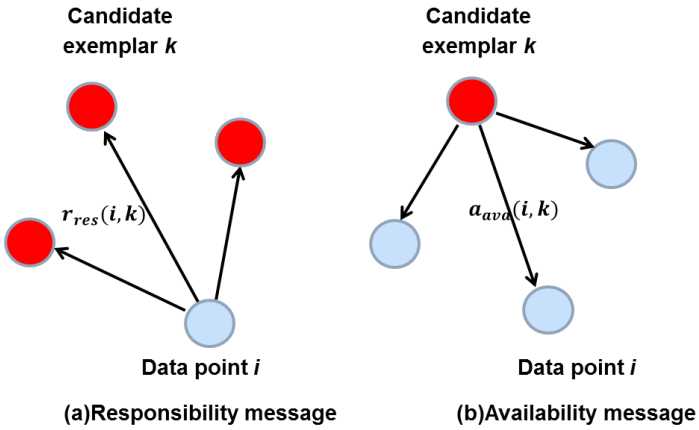


Figure 6-5 (a) Responsibility (b) availability message

The availability message, $a_{ava}(i,k)$, is sent from the candidate exemplar k back to potential cluster member data point i. A candidate exemplar data point k informs other data points whether it is a good exemplar or no, as shown in Figure 6-5(b). The message $a_{ava}(i,k)$ shows how proper it would be for point i to choose point k as its exemplar based on the supporting feedback from other data points. The calculation of the availability considers only the positive responsibility messages from the surrounding data points. If point k receives strong responsibility messages from its surrounding data points, it will send a stronger availability message to indicate its suitability to become an exemplar. Figure 6-6 depicts how the responsibility and availability update messages are exchanged between the data points.

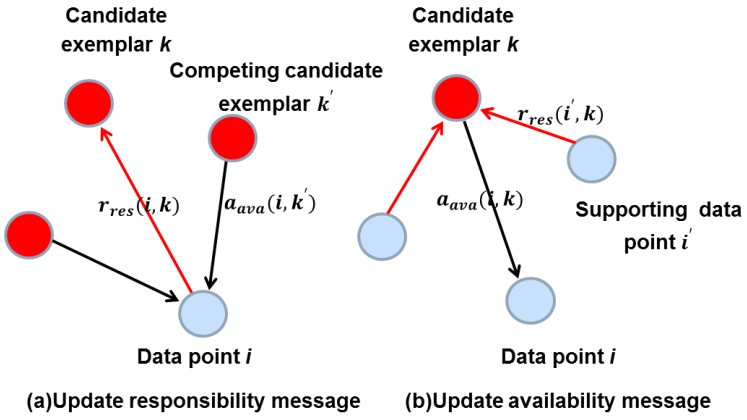


Figure 6-6 (a) Responsibility update (b) availability update message

The responsibilities and availabilities are updated according to the following equations [78]:

$$r_{res}(i,k) = s(i,k) - max_{k':k' \neq k}\{a_{ava}(i,k') + s(i,k')\} \quad (6\text{-}4)$$

$$a_{ava}(i,k) = min\left\{0, r_{res}(k,k) + \sum_{i':i' \neq \{i,k\}} max\{0, r_{res}(i',k)\}\right\} \quad (6\text{-}5)$$

It should be noted that the self-responsibility, $r_{res}(k,k)$ and self-availability, $a_{ava}(k,k)$ are two additional messages calculated for each data point k. Both of these messages give accumulated evidence that the point k is an exemplar, and are used to find the clusters. The self-responsibility message is based on the input preference value and the maximum value of availability message received from the surrounding data points. Whereas, the self-availability message is based on the number of the positive responsibilities messages received and their values. The self-responsibility and self-availability messages are updated according to the following equations [78]:

$$r_{res}(k,k) = s(k,k) - max_{k':k' \neq k}\{s(k,k')\} \quad (6\text{-}6)$$

$$a_{ava}(k,k) = \sum_{i':i' \neq k} max\{0, r_{res}(i',k)\} \quad (6\text{-}7)$$

The algorithm begins by calculating the responsibilities with the availabilities set to 0. While computing the responsibilities and availabilities according to the simple updating equations (6-4) to (6-7), numerical oscillations may occur that prevent the algorithm from converging. So, the responsibilities and availability update messages are damped according to the following equations [78]:

$$R_{res}(t+1) = (1-\lambda)R_{res}(t) + \lambda R_{res}(t-1) \quad (6\text{-}8)$$

$$A_{ava}(t+1) = (1-\lambda)A_{ava}(t) + \lambda A_{ava}(t-1) \quad (6\text{-}9)$$

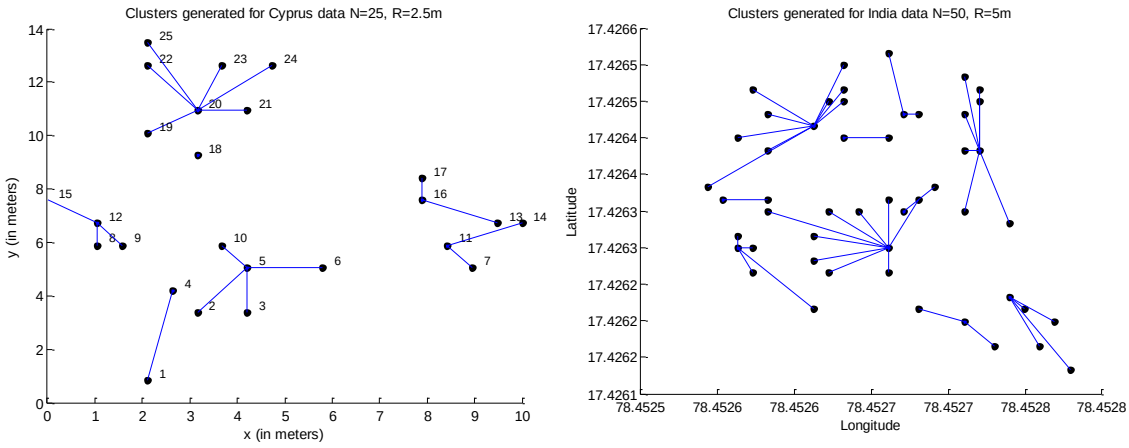
where $R_{res} = [r_{res}(i,k)]$ and $A_{ava} = [a_{ava}(i,k)]$ represent the responsibility matrix and availability matrix respectively. t indicates the iteration times and λ is the damping factor. A value of 0.9 is used as the damping factor λ in this thesis. After the messages have converged, there are two ways to identify the exemplars:

1. For data point i, if $a_{ava}(i,i) + r_{res}(i,i) > 0$, then data point i is an exemplar.

2. For data point i, if $a_{ava}(i,i) + r_{res}(i,i) > a_{ava}(i,j) + r_{res}(i,j)$ for all j not equal to i, then data point i is an exemplar.

This clustering procedure may be performed at any iteration of the algorithm, but the final clustering decisions should be made once the algorithm stabilizes. The algorithm can be terminated once exemplar decisions become constant for some number of iterations, indicating that the algorithm has converged.

The AP clustering has been implemented in MATLAB and was verified using the toy dataset provided by the authors of affinity propagation algorithm [78],[79]. Figure 6-7(a) and (b) show the clusters produced running the AP clustering on the spatially interpolated datasets. The same network topology is used as used for HAC-DC in Figure 6-4. The number of nodes is set to 25 and the transmission radius to 2.5m for the Cyprus datasets while the number of nodes is set to 50 and the transmission radius to 5m for the India datasets. The reason for choosing these network settings are explained in Section 6.5.



(a) (b)

Figure 6-7 Clusters generated using AP clustering for (a) Cyprus (b) India data

The number of clusters generated for the Cyprus dataset is 7, while the number of clusters for the India dataset is 9 using the AP clustering. The number of clusters formed is smaller in case of AP clustering in comparison to the HAC-DC clustering. It can be seen that all the nodes are clustered and no single node clusters are formed in case of AP clustering because of the centralized execution. It can also be noted from Figure 6-7(a) and (b) that the clusters are centred on the exemplars outputted by

the algorithm. For example, in Figure 6-7(a), the exemplars are the nodes 1,5,11,12,16,20.

6.4.4 Data reconstruction

One of the tasks that need to be performed is to estimate/reconstruct the missing data for the non-sampled nodes using the data from the sampled data. This process can be carried out in an offline manner once all the data is available for storage in the database. A reconstruction is also required in order to compute and infer the quality of information gathered by the sampling algorithm. In the current work, mean deviations are used as one of the performance metrics for evaluation of the sampling algorithm. In Section 6.6, more details on the mean deviation performance metric is given. The mean deviations between the real and sampled data are computed for all the nodes after reconstructing the data for the non-sampled nodes. Polynomial regression [32] can be used to derive the relationship between the various nodes using the data at each full sampling cycle. Based on the derived relationships, the data for the non-sampler nodes can be reconstructed from the sampler node data to which it exhibits maximum correlation in the cluster.

Data from the sampler node and the non-sampler node can be denoted by two variables X and Y and their relationship can be reasonably represented by a straight line. This can be represented mathematically as:

$$Y = \alpha X + \beta + \varepsilon \qquad (6\text{-}10)$$

where α describes where the line crosses the y-axis, β describes the slope of the line, and ε is an error term that describes the variation of the real data above and below the line. Simple linear regression attempts to find a straight line that best 'fits' the data, where the variation of the real data above and below the line is minimised.

Figure 6-8 shows the linear regression relationship derived between the data from a non-sampler node (Node 23) and its associated sampler node (Node 21) at the end of a full sampling cycle. The data from both the sampler and non-sampler node are plotted along different axis and the best fit straight line describing the relationship between the two nodes has also been plotted.

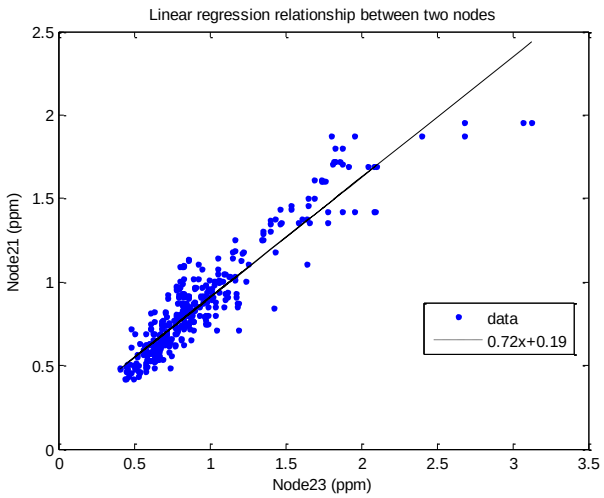


Figure 6-8 Linear regression relationship between sampler and non-sampler nodes

Figure 6-9 shows the reconstructed data for the non-sampler node (Node 23) using the linear regression relationship (0.72x+0.19) and the sampled data collected by the sampler node (Node 21) during one of the adaptive time cycles. It can be observed that the reconstructed data for non-sampler node follows the sampler node's data very closely.

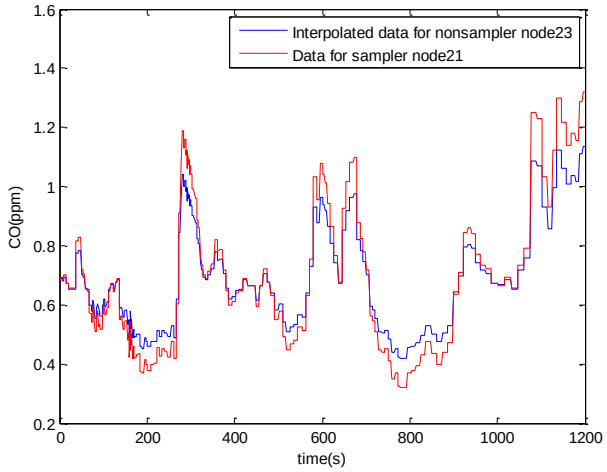


Figure 6-9 Reconstructed time series for a non-sampler node from the sampler node

The section 6.4 provided complete overview of the EDSAS-S algorithm and its operation in details. In the next section, details of the clustering analysis are given and results are presented for both Cyprus and India datasets.

6.5 Clustering analysis

The behaviour of the two clustering algorithms – HAC-DC and AP clustering is studied using the simulated pollution datasets. This study is carried out in order to understand the difference in the clustering behaviour of the two algorithms and also to guide the network settings. The simulations are carried out using the spatially interpolated datasets as explained in Section 6.3. Different network scenarios are generated using random node locations from the spatially interpolated field. The clustering algorithm performance is measured based on the following metrics:

1. Total number of clusters
2. Average cluster sizes (number of nodes per cluster)
3. Total number of rounds for clustering completion
4. Total clustering message overhead

In order to achieve a good sampling performance, the clustering algorithm should yield less number of clusters that are large in size, the number of rounds for clustering completion should be small and the clustering message overhead should be small too. First the clustering performance analysis is presented for HAC-DC in Section 6.5.1 and later for the AP clustering in Section 6.5.2.

6.5.1 Clustering performance of HAC-DC

First, the effect of the correlation threshold on the clustering behaviour is investigated. The number of nodes, N is set to 25 and the transmission radius, R is set to 2m for the Cyprus datasets. For the Indian datasets, N is set to 50 and R is set to 5m. The correlation threshold varies from 0.5 to 0.9 and results are shown for both the Cyprus and the Indian datasets in Figure 6-10(a)-(d). Twenty different simulations are carried out for each experiment to generate different network topologies. The results shown in this section depict the mean across the twenty simulations. The error bars depict the standard deviation obtained across the twenty different simulations and show the effect of the variation of different topologies.

It can be seen from Figure 6-10(a) that the number of clusters increase as the correlation threshold increases for both pollution datasets, since only the very closely located nodes that exhibit high degree of correlations are clustered together. Further

apart nodes exhibit different correlations and hence they form different clusters, consequently this result in an overall higher number of clusters.

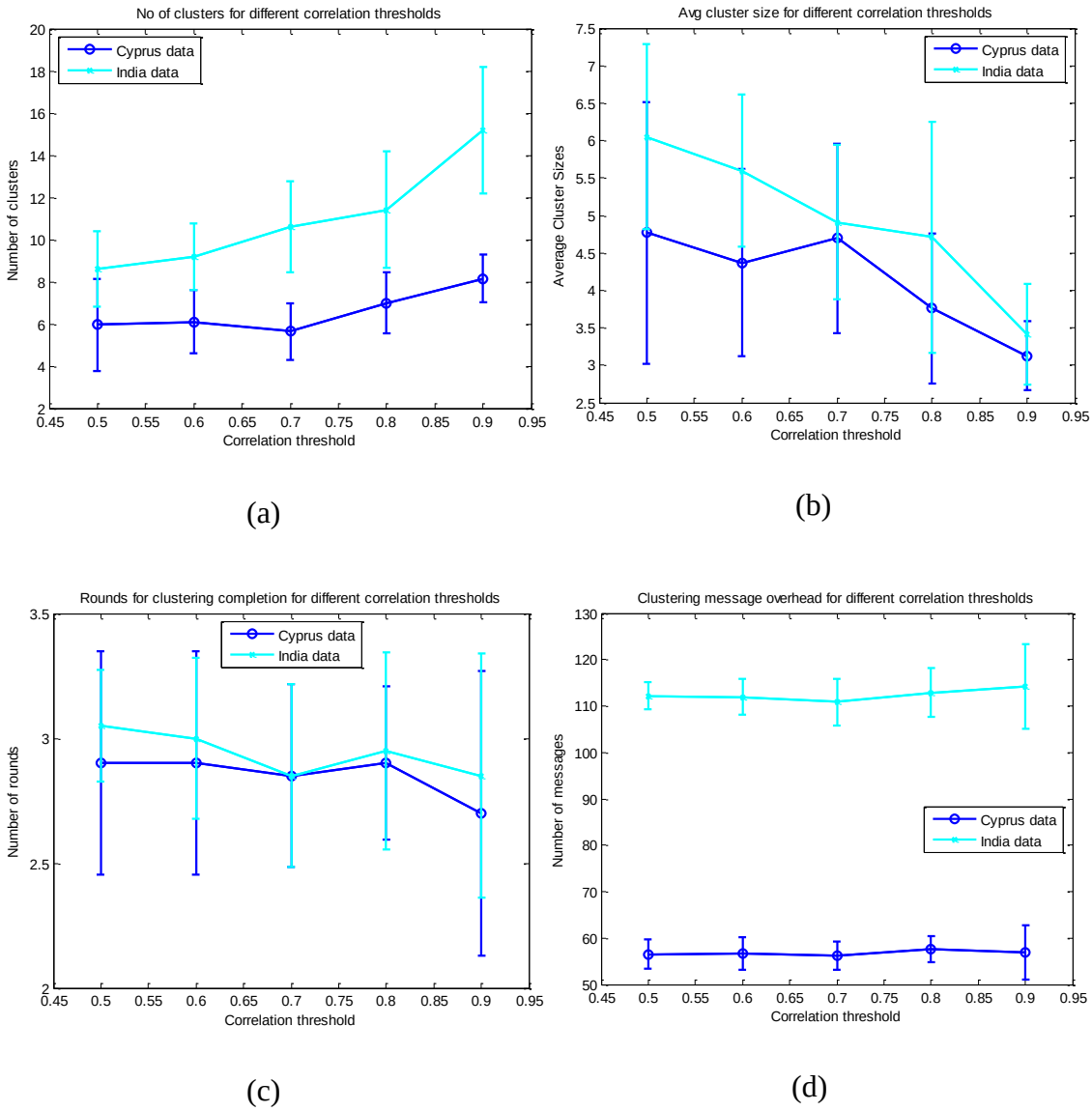


Figure 6-10 Performance results for different correlation threshold values for Cyprus and India datasets using HAC-DC

The average cluster size decreases as the correlation threshold increases (as shown in Figure 6-10(b)), because as the correlation threshold increases, a greater number of small clusters is formed. For small correlation thresholds, there is a higher possibility of cluster merging together and forming larger clusters and hence more nodes will be present in each cluster.

The number of rounds required to complete a clustering is 2 to 4 for both pollution datasets (as shown in Figure 6-10(c)). There is very little change in the

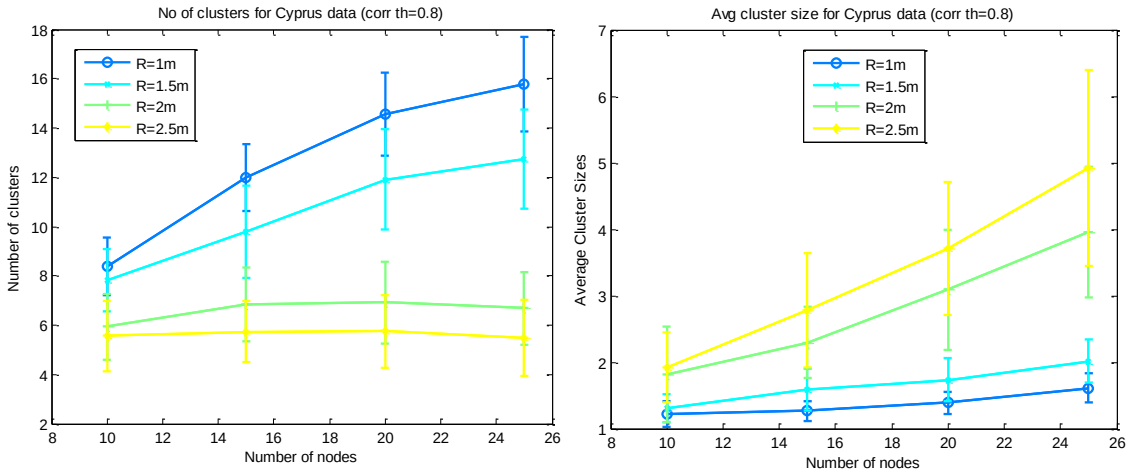
clustering message overhead when changing the correlation thresholds (as shown in Figure 6-10(d)).

Once the correlation threshold is investigated, the influence of the number of nodes (N) and the transmission radius (R) on the clustering performance is evaluated for both the Cyprus and the India pollution datasets. The correlation threshold is set to 0.8 and the time scale is set to 1200s. First the results for the Cyprus datasets are presented. The number of nodes, N varies from 10 to 25 and the transmission radius, R varies from 1m to 2.5m.

Figure 6-11(a) shows that the numbers of clusters follow different patterns for the different transmission radii as the number of nodes increases. For shorter transmission radius (1m and 1.5m), the number of clusters steadily increases as the number of nodes increase. In the case of shorter transmission radius, the nodes do not lie within each other's transmission radius and therefore, more single node clusters are formed. At the longer transmission radii (2m and 2.5m), the number of clusters first increase and then either stay constant or decrease as the number of nodes increase. When there is a higher node density, neighbouring clusters merge together decreasing the number of clusters. Overall, the number of clusters stays below 10 at the longer transmission radii.

In fact, the smaller the number of clusters formed in the network, the better, since it is easier to maintain the clustered architecture and higher sampled data reduction and sensor energy savings can be obtained with larger clusters during the sampling phase, because the fraction of non-sampler nodes increase.

It can be seen from Figure 6-11(b) that as the number of nodes increases, the average cluster size (number of nodes per cluster) increases across all the transmission radii. At longer transmission radii (2m and 2.5m), more nodes can communicate with each other and hence result in larger clusters, whereas at shorter transmission radius (1m and 1.5m), not as many nodes are within each others reach and hence result in smaller clusters.

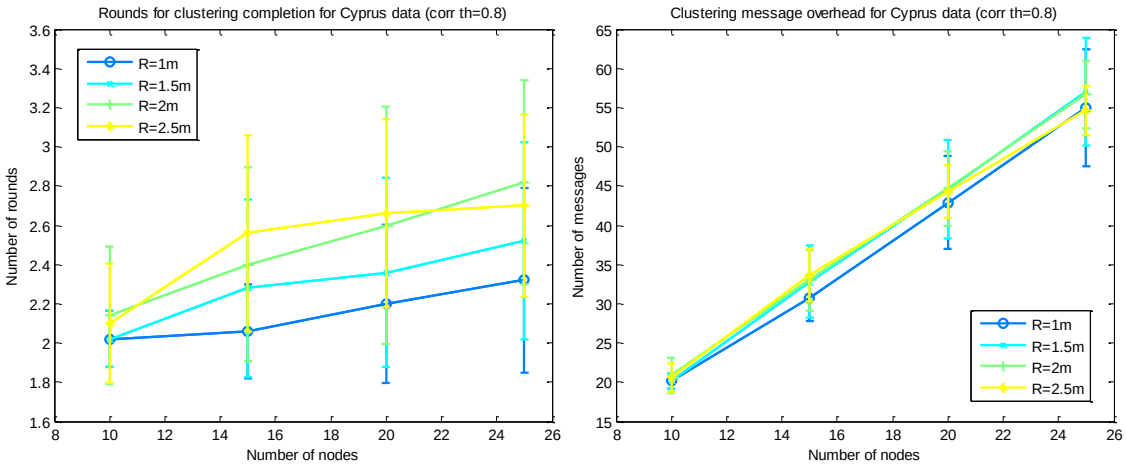


(a) (b)

Figure 6-11 (a) No of clusters (b) average cluster size for Cyprus data for varying nodes and transmission radius using HAC-DC

Figure 6-12(a) shows that the number of rounds for the clustering process increases as the number of nodes increases. This is because more clusters may merge together at subsequent rounds, depending on the spatial correlation patterns. Also, the number of rounds are smaller at shorter transmission radius than at the longer tranmission radius. At shorter transmission radius, the merging of clusters might not happen at the subsequent rounds because they might be out of range of each other, whereas in the case of the longer transmission radius, the merging of clusters still might be possible.

Another observation is that the clustering normally takes an average of 2 to 3 rounds for the Cyprus data, which indicates that the merging of clusters based on the correlation criterion terminates in a finite number of rounds. With 2 to 3 rounds the merging of clusters take place across nodes that are 1 or 2 hops away, and after that the correlations become significantly different. This is an important characteristic of HAC-DC since at every clustering process, only one reorganization of nodes takes place depending upon the changes in the spatial correlation and within a short time span, nodes are able to rearrange themselves into new clusters without incurring any major overheads. This enables HAC-DC to be carried out occassionally at the end of each full adaptive sampling cycle.



(a) (b)

Figure 6-12 (a) No of rounds (b) messaging overhead for Cyprus data for varying nodes and transmission radius using HAC-DC

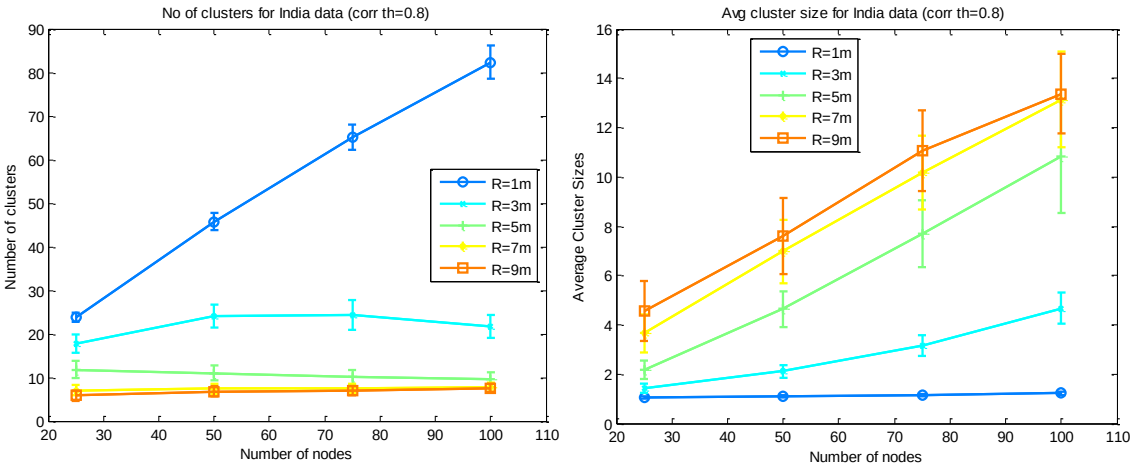
The clustering message overhead in Figure 6-12(b) increases almost linearly as the number of nodes increases, since at each round there are as many hello packets as the number of clusters and as many accept packets as the number of cluster mergings. The transmission radius does not have much impact on the clustering message overhead.

The results for the clustering are obtained for the Indian datasets by varying the number of nodes, N from 25 to 100 and the transmission radius, R from 2 m to 10 m. The correlation threshold is set to 0.8 and the time scale is set to 1200s. The clustering behaviour is shown in Figure 6-13(a)-(b) and Figure 6-14(a)-(b).

It can be seen that the number of clusters follow different patterns for different transmission radius as the number of nodes increases as shown in Figure 6-13(a). For the shorter transmission radii (1m), the number of clusters increases steadily because mostly single node clusters are formed with no nodes being reachable in the neighbourhood. For the longer transmission radii (3m-9m), the number of clusters increase slightly and then stay almost constant afterwards. In fact, for the longer transmission radii (5m-9m), the number of clusters stay below 10.

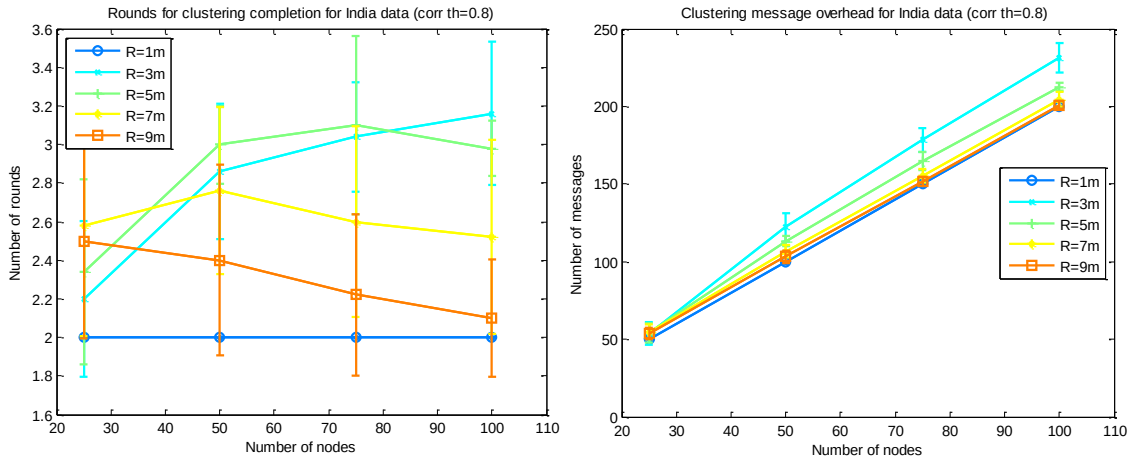
The average cluster sizes increase almost linearly with the number of nodes for the tranmission radii from 3m to 9m as shown in Figure 6-13(b). At the shorter transmission radius, fewer nodes are reachable in one hop distance and hence smaller

clusters are formed in comparison to the longer tranmission radius. Though the average cluster size for the transmission radius of 1m stays almost constant at 1, which indicates no clusters are formed at this transmission radius.



(a) (b)

Figure 6-13 (a) No of clusters (b) average cluster size for India data for varying nodes and transmission radius using HAC-DC



(a) (b)

Figure 6-14 (a) No of rounds (b) messaging overhead for India data for varying nodes and transmission radius using HAC-DC

The number of rounds required for clustering more nodes (75 and 100), is lower for the longer transmission radii (7m and 9m) as shown in Figure 6-14(a), because more nodes are reachable in one hop due to the higher node density and therefore,

nodes are clustered within the intial set of rounds. The tranmisson radius of 1m always takes two rounds since this range is too short for the communication with neighbouring nodes and only single node clusters are formed, so the clustering terminates at the second round.

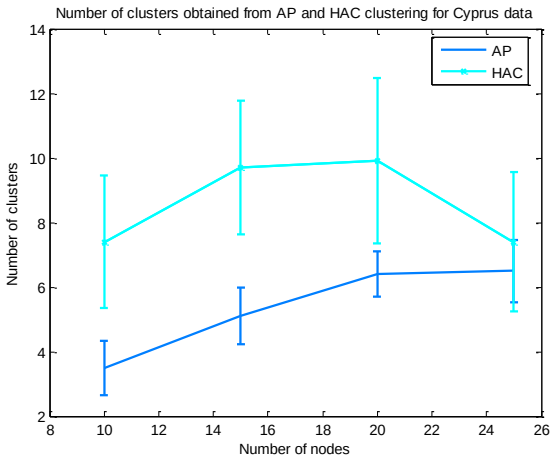
The number of clustering messages in Figure 6-14(b) increases linearly as the number of nodes increases and for the longest transmission radius, the message overhead is lower than the other transmission radii since more nodes can be reached within a single hop and the cluster formation can take place. Therefore, a smaller number of messages are transmitted amongst the nodes.

These set of experiments give an idea about the clustering performance of HAC-DC and it can be seen that the HAC-DC technique gives good performance across the different datasets in terms of lower clustering message overhead (50 to 250 messages), smaller number of clusters formed for longer transmission radius (5 to 10 clusters), smaller number of rounds (2 to 4) for completion for a maximum of 1 to 3 hop clusters. Next the clustering behaviour of AP clustering is analysed.

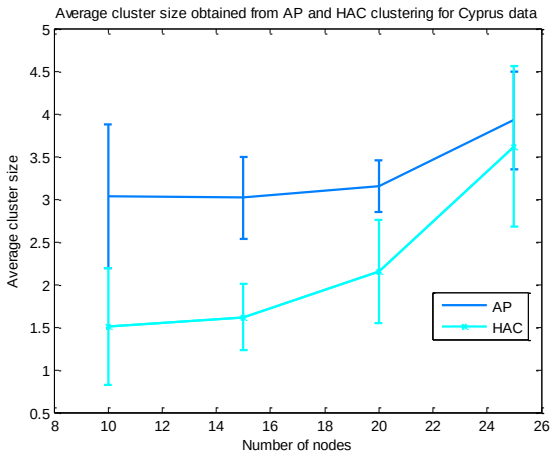
6.5.2 Performance comparison between HAC-DC and AP clustering

This section presents the performance comparison between HAC-DC and AP clustering performance. Figure 6-15(a)-(c) shows the results of the AP clustering for the Cyprus data as the number of nodes, N varies from 10 to 25. Since the AP clustering is performed in a centralized manner, the transmission radius, R parameter is omitted. The initial self-similarity, $s(i,i)$ as explained in Section 6.4.3.3 for all the nodes is set to the median of the correlation distances between different nodes as suggested by the authors of the AP clustering [78]. Ten different simulations are carried out for each experiment to generate different network topologies. The results shown depict the mean across the ten simulations. The error bars depict the standard deviation obtained across the ten different simulations.

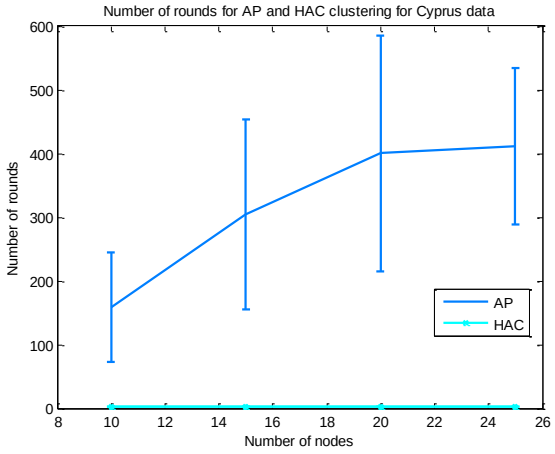
The number of clusters obtained using AP clustering increases as the number of nodes increases as shown in Figure 6-15(a) and the average cluster size also increases as shown in Figure 6-15(b).



(a)



(b)



(c)

Figure 6-15 (a) No of clusters (b) average cluster size (c) no of rounds for Cyprus data for varying nodes using AP clustering

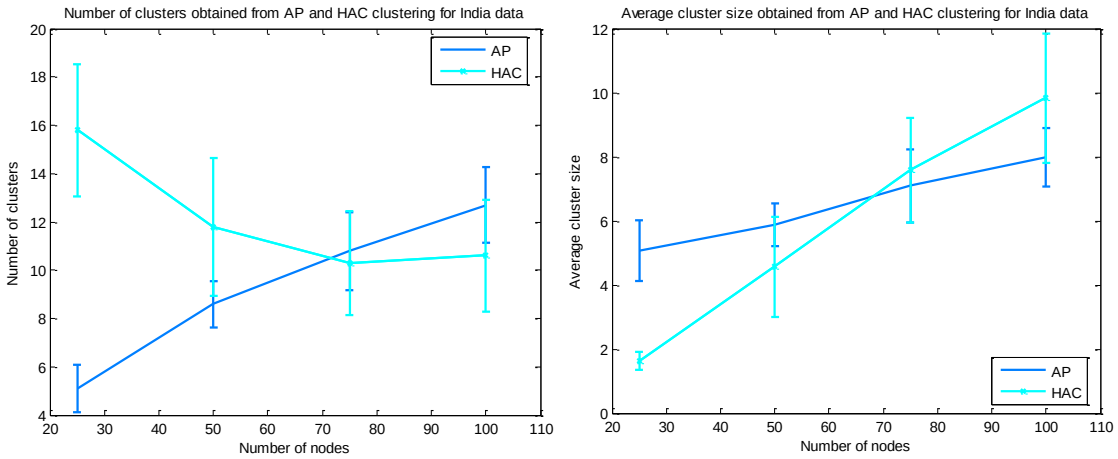
This behaviour of the AP clustering is comparable to the HAC-DC with *R* set to 2m and correlation threshold set to 0.8 (Figure 6-11(a)-(b)). The results for HAC-DC for these network settings are reproduced here again for the sake of comparison. The number of clusters obtained using HAC-DC are higher than those obtained from the AP clustering (as shown in Figure 6-15(a)) and the average cluster size is lower in comparison to that obtained from the AP clustering, as shown in Figure 6-15(b). When the number of nodes reaches 25, the number of clusters and average cluster sizes obtained using HAC-DC and AP clustering overlap with each other for the different topologies. This suggests that the clustering performance is similar for the

two clustering algorithms for these particular network settings. It can be further observed from Figure 6-15(c) that the AP clustering needs more rounds (200 to 400) to converge to the final clusters, which implies a longer execution time and associated delay in the delivery of the final clustering decisions.

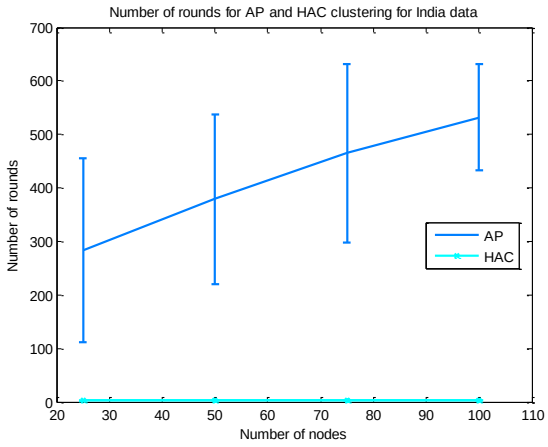
Similarly, Figure 6-16(a)-(c) shows the AP clustering results for the Indian datasets as the number of nodes N increases from 25 to 100. It can be observed that the number of clusters increases as well as the average cluster size increases as the number of nodes increases. This clustering performance is comparable to the one obtained from HAC-DC for the settings with R set to 5m and the correlation threshold set to 0.8 as shown in Figure 6-13(a)–(b).

The number of clusters obtained using HAC-DC is higher for smaller number of nodes in comparison to the AP clustering as shown in Figure 6-16(a). In the case of a smaller number of nodes, more single node clusters are formed, whereas for a larger number of nodes, nearby clusters merge together leading to fewer clusters. It can be seen from Figure 6-16(a) that there is an overlap in the number of clusters when the number of nodes is 50, and as the number of nodes increases from 75 to 100, the overlap first increases and then decreases across the different topologies. This means that when the range of nodes varies between 50 and 100, HAC-DC and AP clustering performance is comparable to each other in terms of the number of formed clusters.

Similarly, Figure 6-16(b) shows that the average cluster size for the Indian data using HAC-DC is smaller at smaller number of nodes and higher at larger number of nodes in comparison to that obtained using the AP clustering. It can be observed that as the number of nodes increase from 50 to 100, there is an overlap in the average cluster sizes obtained from HAC-DC and AP clustering, which means that the performance in terms of average cluster sizes is similar for these particular network settings for the two clustering algorithms. Another point to be noted from Figure 6-16(c) is that the number of rounds (300 to 500) for the AP clustering to converge is very high and increases as the number of nodes increases while the number of rounds needed for HAC-DC is almost negligible in comparison to the AP clustering.



(a) (b)



(c)

Figure 6-16 (a) No of clusters (b) average cluster size (c) no of rounds for India data for varying nodes using AP clustering

The overall insight that can be drawn from these experiments is that HAC-DC gives comparable clustering performance to the AP clustering at particular network settings. For a WSN with few nodes, HAC-DC yields more clusters that are small in size than the AP clustering would yield. Similarly, as the number of nodes increases, the AP clustering yields more clusters that are small in size than HAC-DC. This can be attributed to the HAC-DC distributed execution. However, in case of the AP clustering, the number of rounds required for the clustering convergence is very large and the messaging overhead is higher due to the centralized execution.

6.5.3 Discussion

Two different approaches for clustering nodes on the basis of data correlations have been studied and their clustering behaviour analysed. The AP clustering is able to choose the representative exemplars in a centralized manner and gives good clustering performance in terms of the number of clusters and the average cluster sizes. While in case of HAC-DC, there are no representative cluster heads and instead it recursively compares correlations amongst the nodes in a local neighbourhood to perform a partitioning of the network in a distributed manner.

Based upon the results from the previous sections 6.5.1 and 6.5.2, HAC-DC gives performance comparable to AP clustering in terms of the number of clusters and average cluster sizes for particular network settings. HAC-DC produces slightly more small clusters or less large sized clusters than the AP clustering. This is justified since the communication happens in a limited neighbourhood when compared to the centralized AP clustering. In distributed environments, nodes that are located beyond the transmission range or have significant different correlations will form single node clusters or the clusters that exhibit similar correlations will merge to form larger clusters.

However, when the number of rounds needed for the clustering convergence and the message overhead are taken into account, HAC-DC out performs the AP clustering. The AP clustering is done in a centralized manner so all the data needs to be transferred to the sink in order to compute the clusters and then clustering decisions need to be disseminated to the respective nodes. This incurs a lot of communication overhead in the network. Even in a distributed version of the AP clustering, there will be a large message overhead due to availability and responsibility messages being exchanged between all the pairs of nodes, until they stabilise into the final clusters. In the case of HAC-DC, due to the distributed communication, the clustering messages are exchanged within neighbouring nodes and messages do not need to be sent to the sink. The number of rounds in the case of HAC-DC is limited (2 to 4), which indicates the clustering messaging never goes beyond 1 to 3 hops. Hence the message overhead of HAC-DC is much smaller than the AP clustering.

As per the above analysis, it is justified to choose HAC-DC as the clustering algorithm in this work, since it satisfies the criterion of a simple and a low overhead node clustering mechanism. In Section 6.6.1, both clustering algorithms are incorporated with EDSAS-S and their impact on the sampling performance is evaluated.

6.6 Performance evaluation of EDSAS-S

The performance metrics used to evaluate the performance of EDSAS-S are as follows:

1. *Sampled data reduction*: Sampled data reduction corresponds to the total number of data points that are not sampled by the spatial sampling algorithm. For EDSAS-S, the sampled data reduction is computed for each node using the ratio between the number of non-sampled points and total number of data points in the actual dataset. An average sampled data reduction obtained across all the nodes in the network is reported in the performance results.
2. *Mean deviations*: Mean deviations indicate the quality of data collection and are calculated as the difference between the real and sampled data values across each node in the network. The reconstruction process using linear regression as explained in Section 6.4.4 is carried out for computing the missed data points for the non-sampled nodes before computing the mean deviations. The average percentage of mean deviations is reported in order to estimate the loss of data accuracy across all the nodes in the whole network.
3. *Messaging overhead*: The overall messaging overhead in the network comprises of the total number of communication messages generated by the sampling and clustering mechanisms. The sampling generates the majority of the data packets in the network and can be computed as follows: Say, each run of EDSAS-S yields a total number of sampled points, tsp at each node. Assuming that each data packet to be transmitted to the base station contains n data readings, the total number of sampling messages to be transmitted to the base station is tsp/n. The clustering messages correspond to the control packets that are exchanged among the neighbouring nodes and incurs additional load in the network. Hence, both kinds of messages are accounted for evaluating the EDSAS-S performance. The additional cost caused due to clustering should not override the gain achieved by

using the sampling algorithm. The combined sampling and clustering message overhead metric provides an estimate of the energy consumed in network communication.

Various network simulations are carried out using the spatially interpolated datasets and the above mentioned metrics are evaluated for the different clustering algorithms in Section 6.6.1 and different algorithm parameters like the time scale and the sampling node fraction in Section 6.6.2.

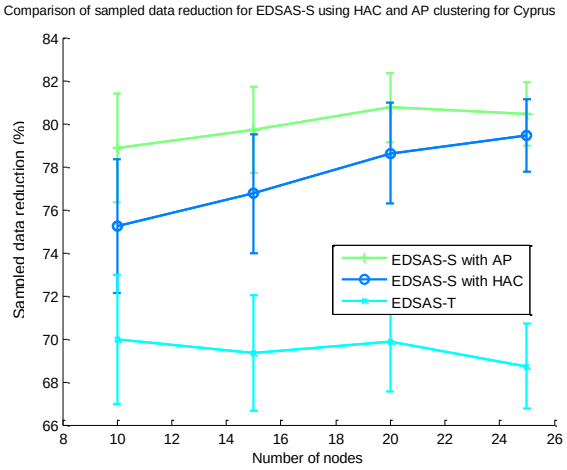
6.6.1 Impact of different clustering algorithms on EDSAS-S performance

In this section, the EDSAS-S is evaluated using the different clustering algorithms and the EDSAS-S performance metrics are computed by varying the number of nodes for each of the pollution datasets. The number of nodes for the Cyprus datasets varies from 10 to 25, while the number of nodes varies from 25 to 100 for the Indian datasets. The time scale is set to 1200s and the correlation threshold is set to 0.9. The transmission radius is set to 2m in case of the Cyprus datasets and 5m in case of the Indian datasets in case of HAC-DC. These network settings are chosen based on the clustering behaviour investigated in the previous Section 6.5. In these experiments only one representative node is chosen for sampling during the adaptive cycle and therefore the algorithm parameter, the sampling node fraction is not used. This is because the AP clustering yields a single exemplar or representative node for each cluster and in case of HAC-DC, the node with maximum remaining energy is chosen as the representative node in each cluster.

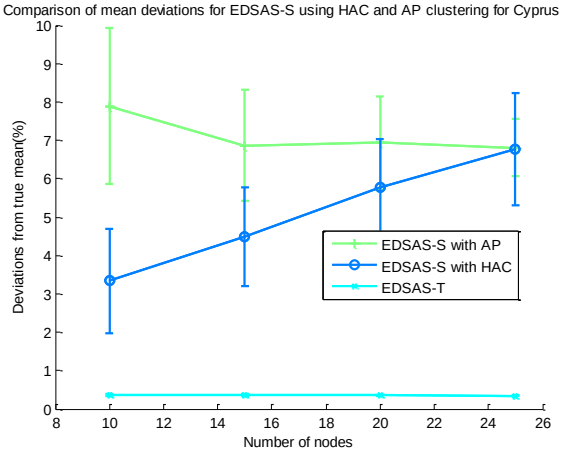
The results shown here represent means across ten different simulations. The error bars depict the standard deviation obtained across the ten different simulations. The results from EDSAS-S are compared for the two different clustering algorithms – HAC-DC and AP clustering and against EDSAS-T, i.e. each sensor node runs and samples data according to EDSAS-T. The measurements using EDSAS-T provides a baseline, since it will provide an indication as to how much benefit is achieved by running the EDSAS-S in comparison to the case when no spatial sampling is used.

First the results for the Cyprus data are shown in Figure 6-17(a)-(c). It can be observed that when the number of nodes is smaller than 25, the data reduction given

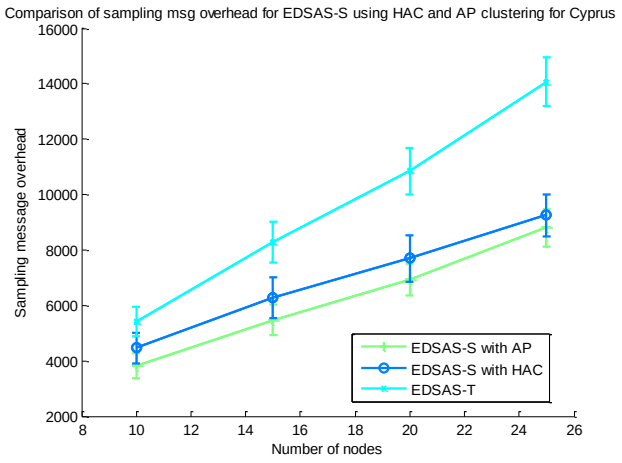
by AP clustering are almost 5% higher than those given by HAC-DC. When the number of nodes increases to 25, the sampled data reduction given by the AP clustering become almost the same (~78%) as those given by the HAC-DC. The reason is that the AP clustering yields smaller number of clusters than HAC-DC leading to smaller number of nodes being chosen as representative nodes in the whole network. The gap between the sampled data reduction using different clustering decrease as the number of nodes increases, since the number of clusters yielded by the AP clustering increases too.



(a)



(b)



(c)

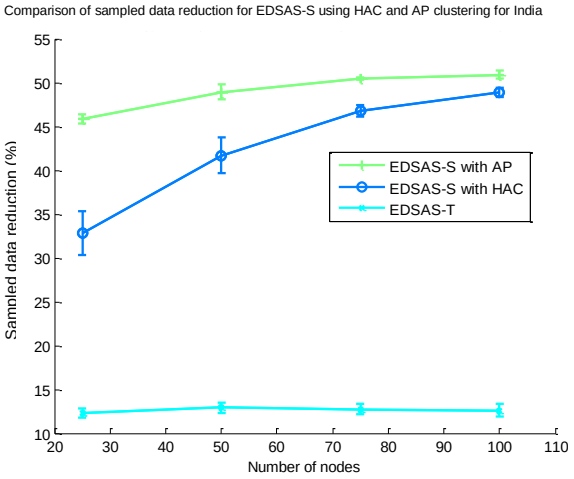
Figure 6-17(a) Sampled data reduction (b) mean deviations (c) sampling message overhead for Cyprus data for varying nodes using different clustering methods

The mean deviations converge for both clustering algorithms when the number of nodes is 25 (~6%), while for smaller number of nodes, the mean deviations for EDSAS-S using the AP clustering are higher (~10%) due to the fewer sampled points in comparison to EDSAS-S using HAC-DC.

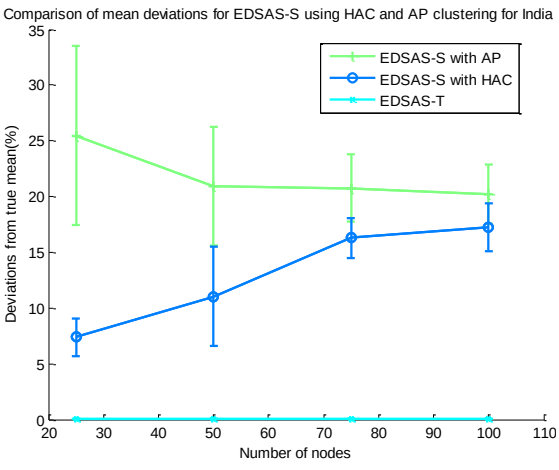
The sampling message overhead using the different clustering algorithms is close to each other though it is slightly smaller in case of EDSAS-S using the AP clustering. This is because in case of EDSAS-S using AP clustering, a smaller number of data points is sampled and therefore, a smaller number of data packets need to be transmitted to the sink.

Next the results for the Indian datasets are shown in Figure 6-18(a)-(c) that are similar in nature to those obtained from the Cyprus datasets. The sampled data reduction in the case of EDSAS-S using HAC-DC is less than EDSAS-S using AP clustering, though at larger number of nodes, the data reduction become almost similar (~45%). When the number of nodes is small, the gap in sampled data reduction due to the different clustering algorithms is slightly higher than 10%.

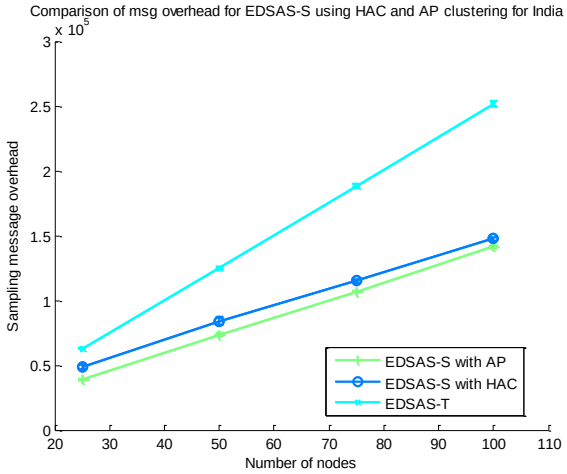
The mean deviations in case of EDSAS-S using AP clustering are quite high (~30%) because of fewer sampled points, but the mean deviations converge with those obtained by EDSAS-S using HAC-DC to around 15% for larger number of nodes. The sampling message overhead yielded using the different clustering algorithms is almost the same.



(a)



(b)



(c)

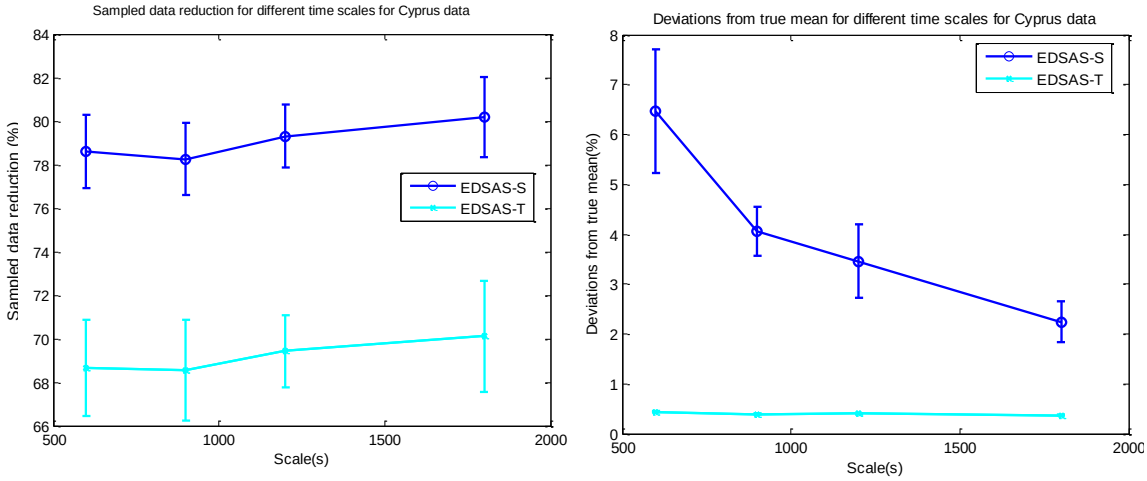
Figure 6-18 (a) Sampled data reduction (b) mean deviations (c) sampling message overhead for India data for varying nodes using different clustering methods

The above results highlight the fact that though the AP clustering provides a good set of clusters and corresponding exemplars to be sampled, EDSAS-S using the AP clustering is able to provide better sampled data reduction, but at a high loss of data accuracy (almost 10% to 30%). However, in case of EDSAS-S using HAC-DC the sampled data reduction is comparatively lower (less than 5% to 10%) than EDSAS-S using AP, but the mean deviations obtained are much lower (5% to 20%) in the case of EDSAS-S using HAC-DC. In fact, for a large number of nodes, the performance of EDSAS-S using HAC-DC becomes similar to that obtained from EDSAS-S using AP clustering. The sampling message overhead using both the clustering mechanisms is almost the same. This study implies that EDSAS-S using HAC-DC sampling performance is as good as EDSAS-S using the AP clustering, while keeping in mind the clustering message overhead that is much lower in the case of HAC-DC clustering, because of the distributed nature of the algorithm in comparison to the centralized AP clustering.

6.6.2 Effect of different time scales and sampling node fractions

In this section, EDSAS-S using HAC-DC is evaluated for the different algorithm parameters - time scale and sampling node fraction for both the Cyprus and the Indian datasets.

Figure 6-19(a)-(b) and Figure 6-20(a)-(b) show the effect of changing the time scale on the various performance metrics for the Cyprus datasets. The various scales used are 600s, 900s, 1200s and 1800s. The number of nodes is fixed to 25, the transmission radius is set to 2m, the correlation threshold is set to 0.8, and the sampling node fraction is set to 0.25. All the results shown in this section represent the mean across ten different simulations. The error bars depict the standard deviation obtained across the ten different simulations. The variation of the scale affects the number of full sampling and adaptive sampling cycles. As the time scale increases, there are less cycles of full and adaptive sampling.



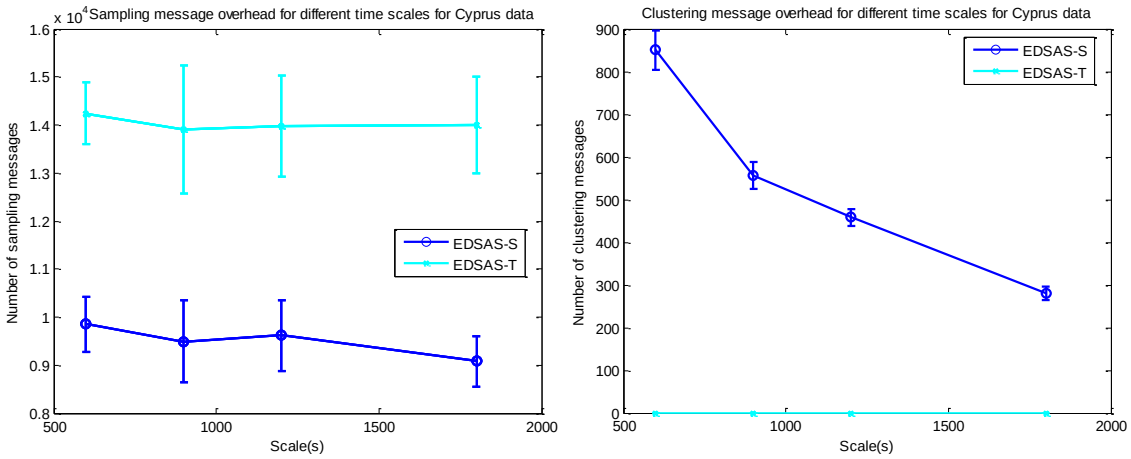
(a) (b)

Figure 6-19 (a) Sampled data reduction (b) mean deviations for Cyprus data for varying time scales

It can be seen from Figure 6-19(a) that the sampled data reduction values given by EDSAS-S always exceed the data reduction values given by EDSAS-T. The difference in the data reduction values is of an order of 15%. Another observation is that the data reduction values stay almost constant as the time scale changes. This shows that the time scale does not have much impact on the sampled data reduction values; this is because the number of data points sampled across different time scales stays almost the same.

EDSAS-S always gives higher mean deviations than EDSAS-T because fewer nodes are sampled in comparison to EDSAS-T. The mean deviations decrease for EDSAS-S as the time scale increases, but always stay below 10%. This is because

the data fluctuations are averaged out during the larger time scales and hence this leads to lower mean deviations.



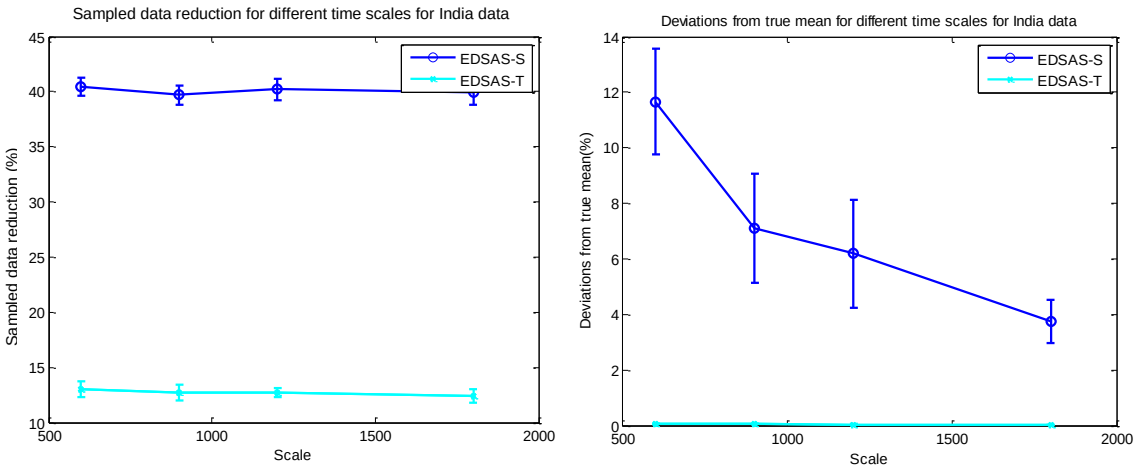
(a) (b)

Figure 6-20 (a) Sampling message overhead (b) clustering message overhead for Cyprus data for varying time scales

EDSAS-S sampling message overhead is always lower than EDSAS-T as shown in Figure 6-20(a) because the non-sampler nodes do not sample any data points in the adaptive cycles in the case of EDSAS-S. This means the energy consumed by EDSAS-S for message communication is lower than EDSAS-T, because of the smaller number of data packets transmitted to the base station. The EDSAS-S sampling message overhead is almost 32% lower than that of EDSAS-T.

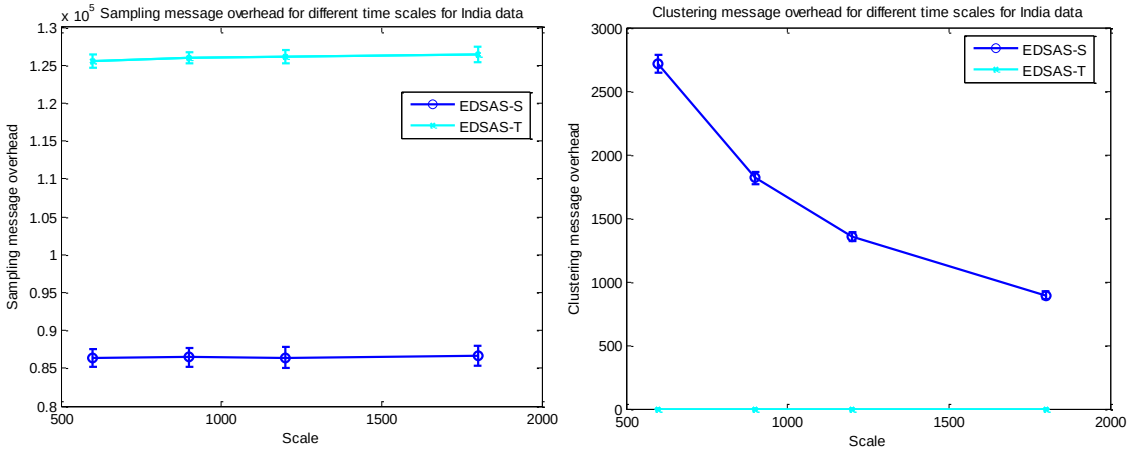
The clustering message overhead (as shown in Figure 6-20(b)) decreases as the time scale increases. This is because the larger time scales results in smaller number of adaptive cycles and hence the clustering algorithm needs to run fewer times on the data. The clustering message overhead for EDSAS-T is negligible since clustering is not used at all in EDSAS-T. It can also be calculated from these graphs that the clustering message overhead is a small percentage (3% to 8%) of the EDSAS-S total (sampling and clustering) message overhead. The savings in the sampling message overhead (24% to 29%) far exceed the clustering message overhead and therefore, clustering proves to be an effective technique for communication energy savings in spatial sampling.

Similarly graphs for the Indian datasets are shown in Figure 6-21(a)-(b) and Figure 6-22(a)-(b). The various scales used are 600s, 900s, 1200s and 1800s. The number of nodes is set to 50, the transmission radius is set to 5m, the correlation threshold is set to 0.8, and the sampling node fraction is set to 0.25.



(a) (b)

Figure 6-21 (a) Sampled data reduction (b) mean deviations for India data for varying time scales



(a) (b)

Figure 6-22 (a) Sampling message overhead (b) clustering message overhead for India data for varying time scales

It can be seen that the sampled data reduction in Figure 6-21(a) is almost constant at different scales. A sampled data reduction of an order of 40% is achieved in the case of EDSAS-S as compared to 15% in the case when only EDSAS-T is used. The

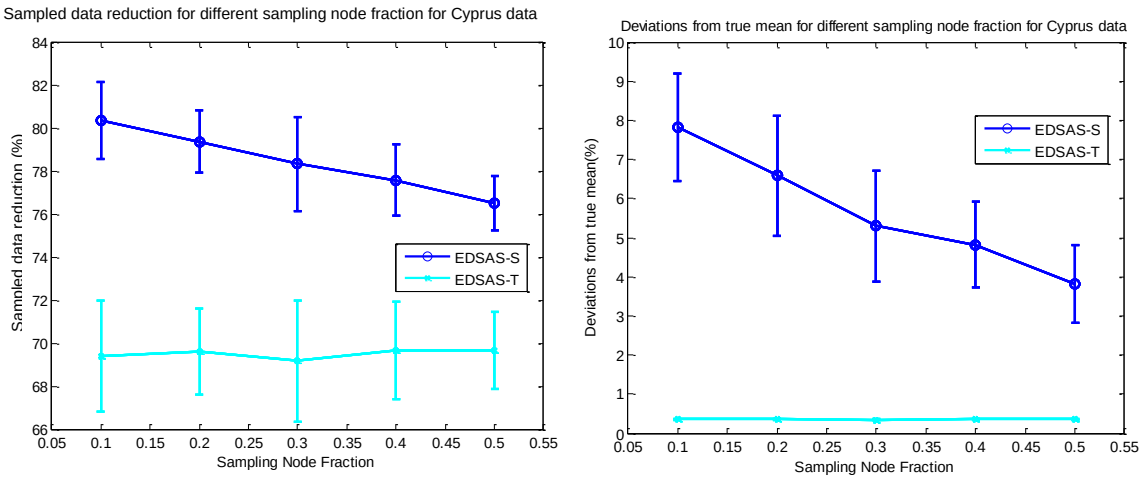
mean deviations again decrease as the scale increases for EDSAS-S as shown in Figure 6-21(b).

The sampling message overhead stays almost constant for different time scales as shown in Figure 6-22(a). The savings in sampling messages due to EDSAS-S is about 32% higher than EDSAS-T. The clustering message overhead (almost 2% of the EDSAS-S total message overhead) decreases as the scale increases since there are fewer number of clustering executions. So, the overall savings in the sampling message overhead is almost 30%.

The sampling node fraction is used to decide the fraction of nodes in a cluster that are designated as the sampler nodes and the remaining cluster nodes are designated as the non-sampler nodes. When varying the sampling node fraction, the number of nodes used for sampling during the adaptive cycles changes, and thus leads to different results for sampled data reduction, mean deviations and messaging overhead. First the results for the Cyprus datasets are shown in Figure 6-23(a)-(b) and Figure 6-24(a)-(b). The number of nodes is set to 25, the transmission radius is set to 2m, the correlation threshold is set to 0.8, and the scale is set to 1200s.

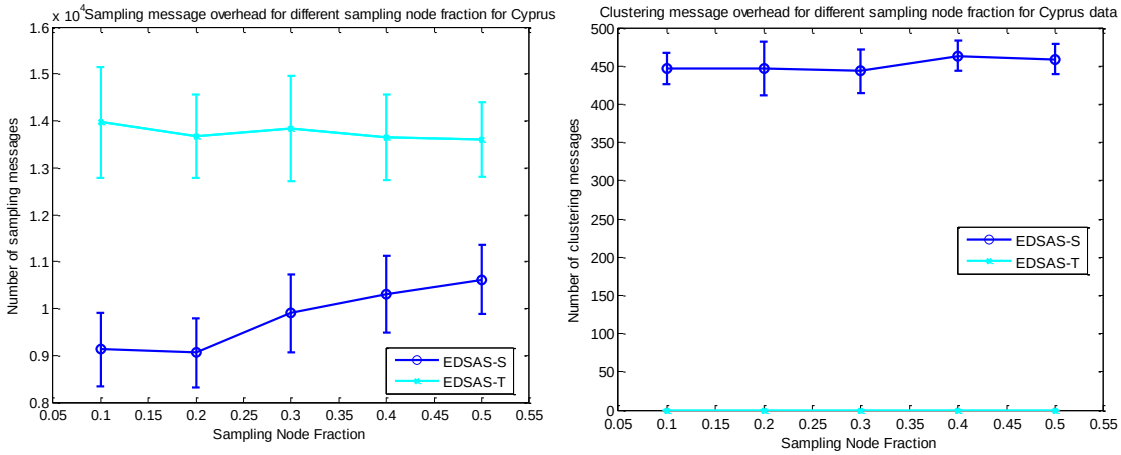
Figure 6-23(a) shows that as the sampling node fraction increases, the sampled data reduction value decreases for EDSAS-S. The mean deviations steadily decrease as the sampling node fraction increases for EDSAS-S (as shown in Figure 6-23(b)). This is very much in line with the expected behaviour since more nodes are used for the sampling in each cluster; the average sampled data reduction values will decrease while yielding better data accuracy.

Figure 6-24(a) shows that the sampling message overhead increases as the sampling node fraction increases for EDSAS-S, because the number of sampling messages generated in the network increases. The overall savings in the sampling message overhead drop from 35% to 28% as the sampling node fraction increases. The clustering message overhead does not change; it constitutes only around 5% of the EDSAS-S total EDSAS-S messaging overhead as shown in Figure 6-24(b). This means an overall 23% to 30% savings in sampling message overhead can be obtained when using EDSAS-S.



(a) (b)

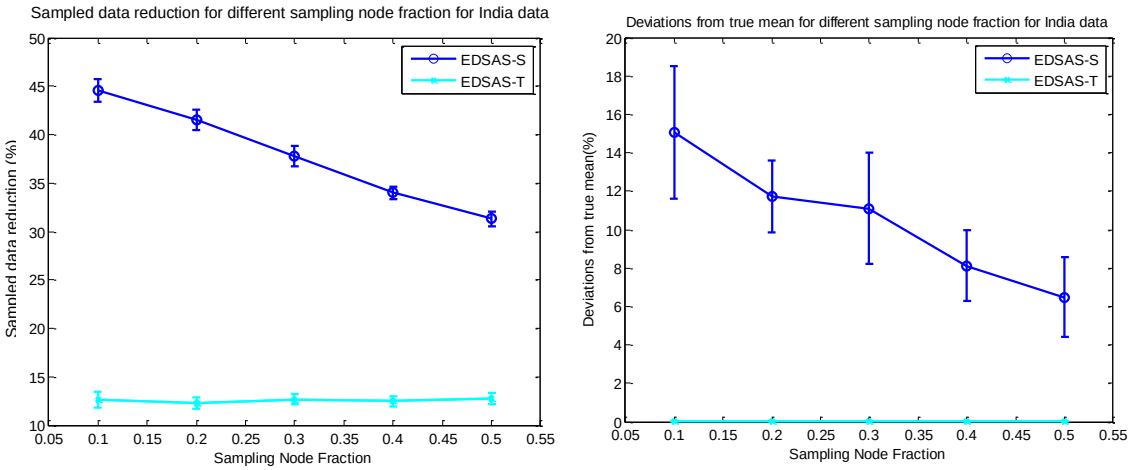
Figure 6-23 (a) Sampled data reduction (b) mean deviations for Cyprus data for varying sampling node fraction



(a) (b)

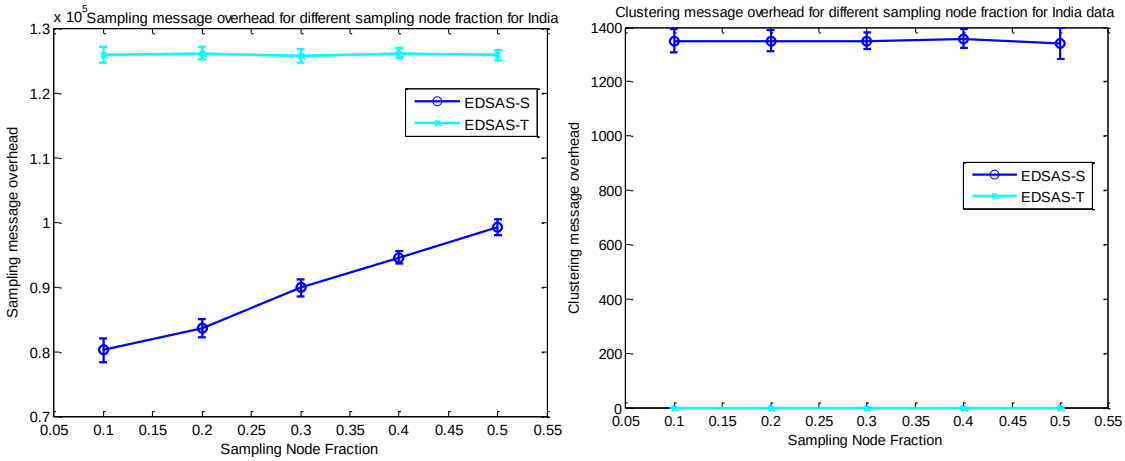
Figure 6-24 (a) Sampling message overhead (b) clustering message overhead for Cyprus data for varying sampling node fraction

Next the results from the Indian datasets are shown in Figure 6-25(a)-(b) and Figure 6-26(a)-(b). The number of nodes is set to 50, the transmission radius is set to 5m, the correlation threshold is set to 0.8, and the time scale is set to 1200s.



(a) (b)

Figure 6-25 (a) Sampled data reduction (b) mean deviations for India data for varying sampling node fraction



(a) (b)

Figure 6-26 (a) Sampling message overhead (b) clustering message overhead for India data for varying sampling node fraction

A decrease in the sampled data reduction takes place as the sampling node fraction increases for EDSAS-S as shown in Figure 6-25(a). The main point is that even though the sampled data reduction decreases as the sampling node fraction increases, data reduction never becomes lower than those obtained from EDSAS-T, i.e. even at a higher percentage of the sampling nodes, the sampled data reduction obtained from EDSAS-S is still higher in comparison to EDSAS-T. EDSAS-T is independent of the change in the sampling node fraction and hence similar results are

obtained for the sampled data reduction. For EDSAS-S, the mean deviations can become very high (almost 15%) at lower sampling node fraction values as shown in Figure 6-25(b).

Figure 6-26(a) shows that the EDSAS-S sampling message overhead increases in these datasets as the sampling node fraction increases. It can be calculated that the savings in the sampling message overhead actually falls from 38% to 23%.The clustering message overhead always constitutes less than 2% of the total message overhead as seen in Figure 6-26 (b). This implies an overall savings of 21% to 36% in the sampling message overhead as the sampling node fraction increases.

This section provided a performance analysis of EDSAS-S across the different pollution datasets for different algorithm parameters. The key observations from these experiments are as follows:

1. EDSAS-S always provides higher sampled data reduction (15% to 25% across different datasets) than only using EDSAS-T on the sensor nodes.
2. The sampling message overhead generated by EDSAS-S is lower than that given by EDSAS-T, which means smaller amount of energy is spent on communicating the data packets to the base station.
3. The clustering message overhead of EDSAS-S is always a small percentage of the total message overhead leading to an overall 20% to 40% higher savings in sampling messages across the different datasets.

6.7 Chapter conclusions

This chapter has provided an explanation about the extension of EDSAS in the spatial domain (EDSAS-S). Most of the spatial sampling algorithms in the research literature do not adapt the sampling rates temporally, but in this thesis, the temporal algorithm is applied in the spatial domain and the benefits are evaluated in terms of the sampled data reduction and the data accuracy. In order to use EDSAS-S, a spatial clustering of nodes on the basis of correlations is introduced as a pre-step to sampling. Hierarchical agglomerative clustering is used, which is a distributed mechanism for clustering nodes with similar readings. It is shown that this clustering algorithm gives good performance in terms of the number of clusters and the average cluster sizes. It takes little time to converge to the final clusters. Its performance is

validated by running and testing against a centralized clustering algorithm called the affinity propagation, which has large associated messaging overhead and takes very long to converge.

The clustering process causes additional communication overhead in the network, but by means of simulations using spatially interpolated data from the pollution trials, it is proven that the additional clustering message overhead produced by HAC-DC is only of an order of 2% to 5%. The total amount of savings accrued in the sampling message overhead is almost 20% to 40% that far exceeds the clustering message overhead and hence justifies the inclusion of an additional clustering process in the spatial algorithm design. Not only that, the sampled data reduction produced by EDSAS-S are of the order of 25% higher for the Indian datasets and 15% higher for the Cyprus datasets than EDSAS-T for a appropriate choice of parameters. In both cases the data accuracy lost is of the order of 10%.

Chapter 7
Nearest neighbours based adaptive spatial sampling

7.1 Introduction

In the previous Chapter 6, an approach for spatial sampling was used in which the nodes were clustered based on the data correlations in the spatial domain, and only a few representative nodes were sampled in order to represent the data from the neighbouring correlated nodes. EDSAS was applied temporally and used to capture the different pollution data characteristics like the local trends and the autocorrelation patterns. Spatial data correlations are an interesting statistical measure of dependence between the different nodes and one of the most commonly used metrics for spatial sampling.

However, there are alternative data interdependence measures that exploit the data characteristics like self-similarity and non-linear dynamics. The nearest neighbour (NN) based predictability measure [39],[40] is one such statistical measure that exploits the non-linear dynamics present in a time series. The data analysis carried out in sections 4.4 and 4.5 proved that the pollution datasets possess self-similarity characteristics and can be modelled using the concepts from non-linear dynamical systems. The NN based predictability measure is used in the design of the spatial sampling algorithm proposed in this chapter, termed as *Nearest Neighbours based Adaptive Spatial Sampling* (NNASS). The NN based predictability measure between different pollution time series gives an idea about the level of inter-dependence relationship between the different nodes. This further gives an insight into which nodes are the best representative nodes in terms of predicting the data for the other nodes and it can also be used to assign adaptive sampling intervals to the remaining nodes. NNASS is also compared against another spatial sampling algorithm called ASAP [85] and the results are presented in this chapter.

The remaining of the Chapter 7 is structured as follows: the details about the nearest neighbour based predictability measure is presented in Section 7.2.1. The spatial sampling algorithm design based upon the NN predictability measure and a detailed example is given in Section 7.3. Section 7.4.1 presents the detailed performance evaluation of the NNASS algorithm for both the Cyprus and the Indian pollution datasets. Section 7.4.1 provides details about both the selection of the default parameter values and the performance results across the different algorithm parameters. Section 7.4.2 presents the performance comparison of NNASS against the ASAP sampling algorithm. A discussion of the limitations and possible extensions of the NNASS sampling technique is included in Section 7.5. Finally Section 7.6 provides the chapter conclusions.

7.2 Nearest neighbour based predictability measure

As explained in the previous section, there is a need to find out the interdependence relationship between the measurements obtained from different nodes. A time series method based on the NN of a phase space representation called time delay embedding [48] can be used to achieve the above stated goal. Embedded vectors provide a generic, high dimensional representation of a dynamical system and can be constructed easily using the historical data. More details about time delay embedding can be found out in Section 3.6.

Due to the presence of self-similarity features in the pollution datasets, the embedded vectors or pieces of time series in the past might have a resemblance to the pieces in the future. The similar patterns of behaviour can be located in terms of NN and can be used to generate data predictions. This kind of application of nonlinear dynamical methods to time series analysis can be used to characterize the amount of predictability in the time series.

Predictability [41] indicates to what extent the past can be used to determine the future. It is a measure of determinism and if the time series is highly deterministic, the neighbouring vectors (representing a similar past) should be followed by close future values. The advantages of this method are that it only uses information local to the points to be predicted and it does not try to fit a function to the whole time series at once. This method can be further extended to *mixed predictability* [42] where interdependence can be detected by assessing the predictability of variables by means

of phase space representation using samples of more than one variable. Assuming that a time series X has a casual dependence on Y, then the additional knowledge of the past of Y should reduce the uncertainty of the future, i.e. improve the predictability of X. The *predictability improvement* of X by Y, denoted by $H(X|Y)$, can be used as a quantitative measure of the *directional dependence* [40],[41],[42] of X on Y. It reflects to what extent the dynamics of X is influenced by the dynamics of Y. Similarly, $H(Y|X)$ can be measured as well and indicates the dependence in the opposite direction.

The method for the NN based predictability measure already exists in the research literature [39],[40] and uses the cross predictions for finding out directional dependencies between the two time series X and Y. The method evaluates how well the embedded vectors of X can anticipate the prediction values of Y and vice-versa and it is depicted in Figure 7-1. The detailed description for the computation of the NN predictability measure is explained in the following sub-section 7.2.1.

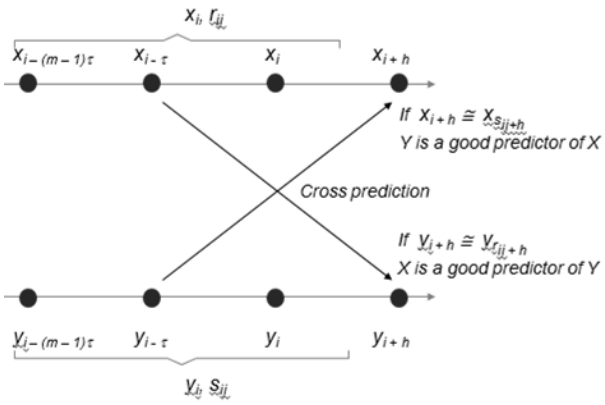


Figure 7-1 Illustration of the cross-prediction using the nearest neighbours [40]

7.2.1 Details about the NN predictability measure

The detailed algorithm for determining the NN based predictability measure between the two time series denoted as, X and Y with measurements x_i and y_i at time instants $i = 1 \dots N$ from two nodes is given as follows [39],[40]. For clarification, an example is considered along with the description of the algorithm using the pollution time series from two different nodes.

Step 1: Find the embedding representation $\boldsymbol{x}_i = [x_i, x_{i-\tau} \dots x_{i-(m-1)\tau}]$ and $\boldsymbol{y}_i = [y_i, y_{i-\tau} \dots y_{i-(m-1)\tau}]$ for both X and Y for N data samples with a given

embedding dimension m and embedding delay τ, where m defines the length of the embedding vectors and τ can be viewed as the time interval between consecutive data samples. The embedded vectors for X can be constructed as follows:

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_{(m-1)\tau+1} \\ \boldsymbol{x}_{(m-1)\tau+2} \\ \vdots \\ \boldsymbol{x}_N \end{pmatrix} = \begin{pmatrix} x_{(m-1)\tau+1} & \cdots & x_1 \\ x_{(m-1)\tau+2} & \cdots & x_2 \\ \vdots & \vdots & \vdots \\ x_N & \cdots & x_{\widetilde{N}} \end{pmatrix} \tag{7-1}$$

where, $\widetilde{N} = N - (m-1)\tau$ is the number of embedded vectors. Similarly the embedded vectors for Y can be constructed as follows:

$$\boldsymbol{Y} = \begin{pmatrix} \boldsymbol{y}_{(m-1)\tau+1} \\ \boldsymbol{y}_{(m-1)\tau+2} \\ \vdots \\ \boldsymbol{y}_N \end{pmatrix} = \begin{pmatrix} y_{(m-1)\tau+1} & \cdots & y_1 \\ y_{(m-1)\tau+2} & \cdots & y_2 \\ \vdots & \vdots & \vdots \\ y_N & \cdots & y_{\widetilde{N}} \end{pmatrix} \tag{7-2}$$

The NN based predictability measure computation is validated for the pollution data taken from two different nodes. Say the two time sequences as shown in Figure 7-2 are X and Y with x_i and y_i , $i = 1 \ldots 20$, $m = 3$ and $\tau = 1$.

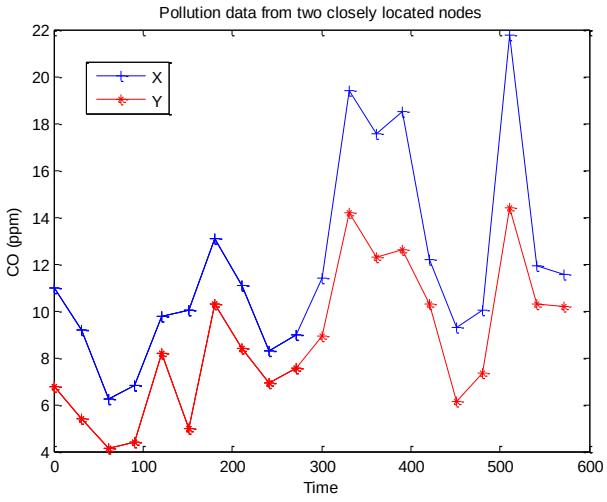


Figure 7-2 Pollution data for NN predictability measure example

The embedded vectors for X and Y are as follows:

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \\ \vdots \\ \boldsymbol{x}_{18} \end{pmatrix} = \begin{pmatrix} x_3 & x_2 & x_1 \\ x_4 & x_3 & x_2 \\ \vdots & \vdots & \vdots \\ x_{18} & x_{17} & x_{16} \end{pmatrix} = \begin{pmatrix} 10.96 & 9.16 & 6.25 \\ 9.16 & 6.25 & 6.8 \\ \vdots & \vdots & \vdots \\ 21.78 & 11.93 & 11.56 \end{pmatrix}$$

$$\boldsymbol{Y} = \begin{pmatrix} \boldsymbol{y}_3 \\ \boldsymbol{y}_4 \\ \vdots \\ \boldsymbol{y}_{18} \end{pmatrix} = \begin{pmatrix} y_3 & y_2 & y_1 \\ y_4 & y_3 & y_2 \\ \vdots & \vdots & \vdots \\ y_{18} & y_{17} & y_{16} \end{pmatrix} = \begin{pmatrix} 6.77 & 5.41 & 4.15 \\ 5.41 & 4.15 & 4.37 \\ \vdots & \vdots & \vdots \\ 14.39 & 10.26 & 10.17 \end{pmatrix}$$

The embedded matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ both will consist of $\widetilde{N}$=18 embedded vectors $[\boldsymbol{x}_3 \dots \dots \boldsymbol{x}_{18}]$ and $[\boldsymbol{y}_3 \dots \dots \boldsymbol{y}_{18}]$ respectively.

Step 2: Find the K NN for each of the embedding vectors $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ say, $p_{i,j}$ and $q_{i,j}$, where $j = 1 \dots K$. K denotes the number of nearest neighbours to be evaluated. The NN of $\boldsymbol{x}_i$ are defined as the embedded vectors $\boldsymbol{x}_j$ that have the smallest values of Euclidean norm $d_{i,j} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$.

For $K = 8$, the K NN of the embedded vector $\boldsymbol{x}_3$ are computed as $\boldsymbol{x_3}, \boldsymbol{x_{10}}, \boldsymbol{x_4}, \boldsymbol{x_9}, \boldsymbol{x_{17}}, \boldsymbol{x_6}, \boldsymbol{x_{11}}, \boldsymbol{x_8}$ and K NN of the embedded vector $\boldsymbol{y}_3$ are computed as $\boldsymbol{y_3}, \boldsymbol{y_4}, \boldsymbol{y_6}, \boldsymbol{y_{10}}, \boldsymbol{y_{17}}, \boldsymbol{y_5}, \boldsymbol{y_{11}}, \boldsymbol{y_9}$ from the pollution data. So, the values for $p_{3,1\dots8}$ are $(3,10,4,9,17,6,11,8)$ and the values for $q_{3,1\dots8}$ are $(3, 4,6,10,5,11,9)$. Similarly, the K NN for all the embedded vectors can be computed using the pollution time series.

Step 3: Calculate a future prediction value for each of the embedded vector $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ say, x_{i+h} and y_{i+h}, where h is the prediction horizon. A prediction model is employed by taking the mean across the values of the K NN embedding vectors.

The prediction value for the embedded vector $\boldsymbol{x}_3$, x_{3+h} is computed as a mean across (x_3, x_2, x_1) and the one step ahead predicted value x_{3+h} is 9.59 and the prediction value for the embedded vector $\boldsymbol{y}_3$ is computed as a mean across (y_3, y_2, y_1) and the predicted value y_{3+h} is computed as 6.67. Similarly, the prediction values for all the embedded vectors can be computed.

Step 4: Compute the differences between the prediction values of each time instance i and the prediction values of the NN $q_{i,j}$ of $\boldsymbol{y}_i$ leading to a distance measure as follows:

$$D_i(X|Y) = \frac{1}{K}\sum_{j=1}^{K} \left| x_{i+h} - x_{q_{i,j+h}} \right| \qquad (7\text{-}3)$$

where, j is the index of the jth NN and x_{i+h} is the prediction value of x assigned to $\boldsymbol{y}_i$ and $x_{q_{i,j+h}}$ is the prediction value assigned to the jth NN of $\boldsymbol{y}_i$. In this method, the prediction value y_{i+h} is assigned to the embedded vector $\boldsymbol{x}_i$ instead of being assigned to $\boldsymbol{y}_i$ and similarly the prediction value x_{i+h} is assigned to the embedded vector $\boldsymbol{y}_i$.This cross prediction method is also shown in Figure 7-1. For robustness in the presence of outliers, the measure is summed over the K nearest neighbours instead of choosing just one single nearest neighbour. Outliers can distort the result by accidentally being a particularly good or poor predictor, and without averaging over the K nearest neighbours, the impact of an outlier would be significant.

The distance measure $D_3(X|Y)$ is computed by taking a summation across the difference between the predicted values x_{3+h} assigned to $\boldsymbol{y}_3$and prediction values of the NN of $\boldsymbol{y}_3$, x_{3+h} , $x_{4+h}, x_{6+h}, x_{10+h}, x_{17+h}, x_{5+h}, x_{11+h}, x_{9+h}$. In this case, the value of $D_3(X|Y)$ is 0.31. Similarly, the distance measure is computed across all the embedded vectors. If the difference between the prediction values is large across all the embedded vectors, i.e. the measure $D(X|Y)$ is large, then Y is a poor predictor of X .

Step 5: Similarly compute a self-predictability measure for all the time instances i, as follows:

$$D_i(X) = \frac{1}{K}\sum_{j=1}^{K}\left|x_{i+h} - x_{p_{i,j+h}}\right| \tag{7-4}$$

where, x_{i+h} is the prediction value of x assigned to $\boldsymbol{x}_i$ and $x_{p_{i,j+h}}$ is the prediction value assigned to the jth NN of $\boldsymbol{x}_i$.

The distance measure $D_3(X)$ is computed by taking a summation across the difference between the predicted values x_{3+h} and prediction values of the NN of $\boldsymbol{x}_3$, x_{3+h} , $x_{10+h}, x_{4+h}, x_{9+h}, x_{17+h}, x_{6+h}, x_{11+h}, x_{8+h}$. In this case, the value of $D_3(X)$is 0.31. Similarly, the distance measure is computed across all the embedded vectors.

Step 6: Compute the accumulated interdependence measure from the distance measures calculated for all the embedded vectors and time instances i as follows:

$$H(X|Y) = \frac{1}{\widetilde{N}} \sum_{i=1}^{\widetilde{N}} \frac{D_i(X|Y)}{D_i(X)} \tag{7-5}$$

Similarly, $H(Y|X)$ can also be computed using the steps 5-7 by exchanging X and Y. If $H(X|Y)$ is small, it indicates that Y is a better predictor of X and if $H(X|Y)$ is large, then Y is a poor predictor. This is an important feature of the predictability measure and is used in the NNASS algorithm design in Section 7.3.

Finally, $H(X|Y)$ is computed once the distance measures $D(X|Y)$ and $D(X)$ are computed across all the embedded vectors. The values of $H(X|Y)$ obtained is 1.21. The value of $H(Y|X)$ is 1.03. The results imply that X is a better predictor of Y for the example data shown in Figure 7-2.

7.2.2 Verification of the NN based predictability measure

The verification of the NN based predictability measure is done in order to establish the correctness of the MATLAB implementation. The NN based predictability measure is verified using two uni-directionally coupled Henon maps. Henon map are a discrete time dynamical system commonly studied in order to understand the properties of non-linear dynamical systems. Following equations (7-6)-(7-7) define the driver system and equations (7-8)-(7-9) define the response system[49],[50]:

$$x_1{}' = 1.4 - x_1^2 + b_x x_2 \tag{7-6}$$

$$x_2{}' = x_1 \tag{7-7}$$

$$y_1{}' = 1.4 - (C x_1 y_1 + (1 - C) y_1^2) + b_y y_2 \tag{7-8}$$

$$y_2{}' = y_1 \tag{7-9}$$

where, C denotes the coupling strength or the interdependence between the two Henon maps. The predictability measures $H(X|Y)$ and $H(Y|X)$ are evaluated with b_x, b_y set to 0.3, m set to 3, K set to 20 for different amount of coupling strengths. It can be seen that at the lower coupling strengths, $H(X|Y)$ is greater than $H(Y|X)$ indicating that Y is a bad predictor of X since the degree of dependence of Y on X is smaller. As the coupling strength between the two Henon maps increases, $H(X|Y)$

and $H(Y|X)$ converge with each other for C greater than 0.7 and both X and Y can predict each other well. This is the expected behaviour from these types of non-linear dynamical systems. This verifies that the results for the NN predictability measurement are in conformance to the expected behaviour.

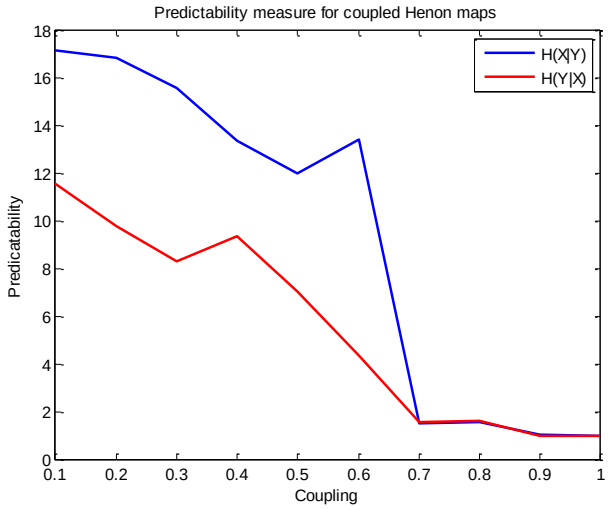


Figure 7-3 H(X|Y) and H(Y|X) for two coupled Henon maps

7.2.3 Significance analysis of the NN based predictability measure

A threshold value is needed to determine whether a value of $H(X|Y)$ is significant or not. If the value is not significant, then an interdependence relationship between the two time series X and Y can be ruled out. The method of surrogate time series can be used to perform the significance analysis for the predictability measure. More details about the surrogate time series can be found in [48],[51].

A surrogate time series has a power spectrum, an autocorrelation function, and a probability density function that are well matched to those of the time series under test, but with the phases of the frequency components randomized. The amplitude adjusted Fourier transform (AAFT) method can be used to create the surrogate time series. The key steps for creating the surrogate series using the AAFT method are as follows[51]:

1. The original data values are re-scaled to convert the amplitude distribution to Gaussian

2. Random phases are added to the arguments of the discrete Fourier transform (DFT), and

3. A re-inversion via the inverse DFT is carried out and a re-scaling is done in order to restore the original amplitude distribution.

Such surrogate time series derived from *X* and *Y* have no interdependence because of the phase randomization, but they retain the key aspects of the original time series that influence the choice of the embedding parameters. Figure 7-4 shows the five different surrogate time series obtained using the AAFT method for one of the nodes from the Cyprus trials, Node7 (original data for a single time scale of 1200s is shown 9eeewsin the topmost subplot).

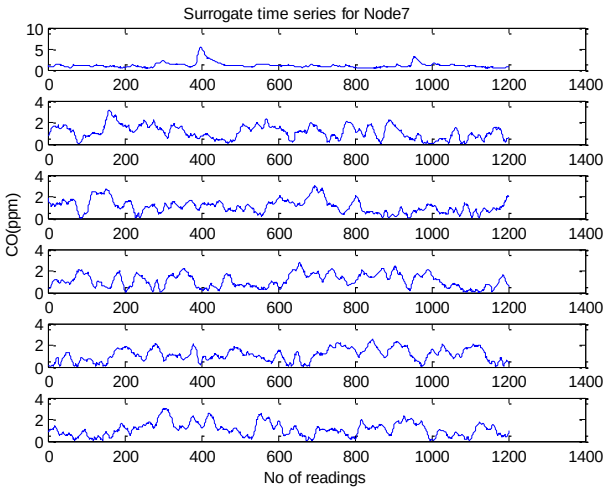


Figure 7-4 Surrogate time series for Node7

In order to carry out the significance analysis, a null hypothesis is chosen such that the time series X does *not* influence the time series Y and the predictability measure between X and Y is denoted by $\lambda_0 = H(X|Y)$. For each of the time series X and Y, a number of surrogate time series are created defined by X_k and $Y_k, k = 1 \ldots\ldots N_{surr}$. N_{surr} defines the total number of surrogate time series generated. For the experiments in this work, N_{surr} is set to 20. Once the N_{surr} number of surrogate time series are generated, the predictability measure values are computed for each of the surrogate time series and defined as $\lambda_k = H(X_k|Y_k)$ for $k = 1 \ldots\ldots N_{surr}$.

The null hypothesis is rejected if the test statistic λ_0 varies considerably from the statistic λ_k generated for all the surrogate values. Let μ_λ and σ_λ denote the sample mean and standard deviation of the distribution of λ_k. The measure of "significance"

is defined as the difference between the original statistic and the mean of the surrogate statistic values, divided by the standard deviation of the surrogate statistic values. In this case the statistic used is the predictability measure. The significance level is estimated to measure the deviation of the predictability measure of the original from the surrogate as follows [51]:

$$\zeta = \frac{|\lambda_0 - \mu_\lambda|}{\sigma_\lambda} \tag{7-10}$$

with the mean value computed as follows:

$$\mu_\lambda = \frac{1}{N_{surr}} \sum_{k=1}^{N_{surr}} \lambda_k \tag{7-11}$$

and the variance computed as follows :

$$\sigma_\lambda{}^2 = \frac{1}{N_{surr} - 1} \sum_{k=1}^{N_{surr}} (\lambda_k - \mu_\lambda)^2 \tag{7-12}$$

and the p-value is generated as follows [51]:

$$p = erfc(\zeta/\sqrt{2}) \tag{7-13}$$

where erfc is the complementary error function [52] defined as follows:

$$erfc(x) = \frac{2}{\sqrt{\pi}} \int_x^{-\infty} e^{-t^2} \, \mathrm{dt} \tag{7-14}$$

The p-value is the probability of observing a significance ζ or larger if the null hypothesis is true. If $p \leq 0.05$, then the null hypothesis that X does not influence Y can be rejected, i.e. the interdependence relationship between X and Y is significant. In the Section 7.3, it is explained how the NN based predictability measure and its associated significance level is used for spatial sampling in pollution sensor networks.

7.3 Detailed description of the NNASS algorithm

Based on the NN predictability measure, the spatial sampling algorithm is proposed in this thesis and it is termed as Nearest Neighbours based Adaptive Spatial Sampling (NNASS). In the algorithm, the sensing schedule of each node consists of

few $full_{sense}$ cycles and few $adapt_{sense}$ cycles each of time duration defined by a time scale s. In each $full_{sense}$ cycle, the nodes sample at a pre-defined sampling interval determined by the embedding delay value τ. In each $adapt_{sense}$ cycle, the nodes sample data at intervals either equal to the embedding delay or an adapted sampling interval. The NNASS sampling strategy is shown in Figure 7-5.

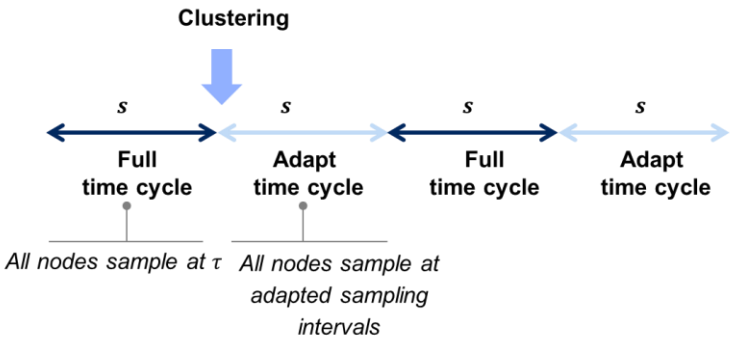


Figure 7-5 Sampling strategy of NNASS

It should be kept in mind that in the case of NNASS, the base sampling interval at which a node can sample is specified by the embedding delay value. The appropriate time scale s and embedding delay τ can be pre-determined by means of an offline data analysis for pollution datasets and is explained in the next Section 7.4.1 on NNASS algorithm parameter selection.

The NNASS algorithm will run in a clustered network architecture and a node clustering algorithm needs to be executed prior to running the spatial sampling algorithm. The idea is that the node clustering algorithm should not be run too frequently and the spatial node clusters created will stay intact for a given time duration. For example: It can be run at different time of the day to reorganize the clustering depending to the traffic levels and meteorological conditions. Hierarchical agglomerative clustering based on data correlations (HAC-DC) as specified in Section 6.4.3.2 is used to generate the node clusters in this algorithm. This clustering mechanism will ensure that the network is partitioned into groups of nodes with similar data readings.

Now say, for each cluster C, there are $n(C)$ cluster nodes denoted by C_i, $i = 1 \ldots . n(C)$. Within each cluster C, the various steps involved in the NNASS algorithm are described as follows:

Step 1: During the $full_{sense}$ time cycle, each node collects data for the pre-defined time scale s at a base sampling interval equal to the chosen embedding delay τ.

Step 2: At the end of the $full_{sense}$ time cycle, the sensed data is collected at one of the designated cluster nodes. One way is to choose a particular cluster node for NN based predictability measure calculations and then rotate the responsibility amongst different cluster nodes after every $full_{sense}$ time cycle so as to provide computation load sharing amongst the cluster nodes. Another way is to carry out the NN based predictability measure calculations at a dedicated resourceful node since it is a computationally intensive task.

Step 3: The NN based predictability measure algorithm as described in Section 7.2.1 runs inside each cluster and the predictability measure between each pair of nodes is computed using the chosen embedding parameters. A predictability matrix showing the interdependence measure, $H(C_i|C_j)$ for all the pairs of nodes, $C_i, C_j, i = 1 \ldots . n(C), j = 1 \ldots . n(C)$ in the cluster C is generated. The corresponding significance level of the interdependence measures for all the pairs of nodes is also evaluated using the method given in Section 7.2.3 and all the pair of nodes for which the interdependence measure is insignificant, $H(C_i|C_j)$ values are set to zero. As a result, the predictability matrix will contain only the significant interdependence values amongst any given pairs of nodes. For each cluster node C_j, the set of n nodes, $n \leq n(C)$, that possesses a significant interdependence relationship can be represented by C_k where $k = 1 \ldots . . n, C_k \subseteq C_i$

Step 4: Using the predictability matrix, an average $H_{avg}(C_k|C_j)$ value is computed for each node C_j with the remaining cluster nodes C_k. The node with the lowest average predictability measure is sampled at the highest frequency since it is the best predictor node. The sampling rate for the best predictor node is set to the base sampling interval and it is equal to the embedding delay τ value used in the $full_{sense}$ time cycle. The remaining nodes are assigned lower sampling frequencies depending on the computed predictability measure values. The adapted sampling interval for cluster member C_j other than the best predictor node, say t_j is assigned using the following relationship:

$$t_j \sim H_{avg}(C_k|C_j) * \tau \qquad if\ H_{avg}(C_k|C_j) \geq 1$$
$$t_j \sim \tau \qquad if\ H_{avg}(C_k|C_j) = 0 \tag{7-15}$$

The adapted sampling intervals, t_j will be larger than the base sampling interval (τ) because $H_{avg}(C_k|C_j) > 1$. It might be possible that owing to the prevailing data dynamics, there are no significant interdependence relationships between the cluster nodes and $H_{avg}(C_k|C_j) = 0$; in that case the cluster nodes will be assigned the base sampling rate equal to the embedding delay. This implies that the nodes that measure data significantly different from other nodes in the cluster and depict no interdependence whatsoever, they should be sampled at the base sampling rate as well.

Step 5: During the $adapt_{sense}$ time cycle, the node that is the best predictor for the remaining nodes will keep sampling at the highest rate (base/smallest sampling interval) while the remaining nodes will sample at a lower rate (larger adapted sampling intervals) for time scale, s. At the end of the $adapt_{sense}$ time cycle, the algorithm reinitiates from step 1.

The above presented NNASS algorithm is illustrated by means of a numerical example as follows. Let's consider a cluster C with four nodes C_1, C_2, C_3, C_4. The embedding dimension used is 3, the embedding delay (base sampling interval) is 10s, the time scale is 1200s, the number of nearest neighbours is 15, and the prediction horizon is 1. NNASS runs over 12000s of data and hence will consist of five $full_{sense}$ and $adapt_{sense}$ time cycles respectively.

The predictability matrices (denoted by A.1,B.1,C.1,D.1 and E.1) generated along with the significance level matrices (denoted by A.2,B.2,C.2,D.2 and E.2) for each of the five consecutive $full_{sense}$ time cycle are shown in Figure 7-6. Each matrix cell in the predictability matrix shows the $H(C_i|C_j)$ values indicating how much the data from the node C_j on the vertical axis (column j) can help predict the data for the node C_i on the horizontal axis (row i). The highlighted values in the significance level matrices indicate insignificant levels of interdependence between the pairs of nodes (p-value > 0.05) and therefore, the corresponding $H(C_i|C_j)$ values are set to zero in the corresponding predictability matrices. It should be noted that the

diagonal elements in the predictability matrices are 1.0 indicating that the self-predictability measure is 1 and each node is the best predictor for its own self.

A.1) Hxy for full time cycle 1

	C1	C2	C3	C4
C1	1.0000	0	0	4.8500
C2	0	1.0000	0	0
C3	3.7046	0	1.0000	1.2135
C4	3.7881	0	1.1535	1.0000
Mean Hxy	2.8309	1.0000	1.0768	2.3545
Sampling Intervals	30	10	10	20

A.2) Significance Level for full time cycle 1

	C1	C2	C3	C4
C1	3.0266	1.6429	1.8367	3.2287
C2	1.1384	3.6943	0.2800	0.8784
C3	3.2879	0.8991	3.8507	4.1789
C4	4.0178	0.3963	7.3775	6.2509

B.1) Hxy for full time cycle 2

	C1	C2	C3	C4
C1	1.0000	1.8580	0	4.1065
C2	1.8393	1.0000	0	0
C3	0	1.7040	1.0000	1.1319
C4	0	0	1.0957	1.0000
Mean Hxy	1.4197	1.5206	1.0479	2.0795
Sampling Intervals	10	20	10	20

B.2) Significance Level for full time cycle 2

	C1	C2	C3	C4
C1	5.8148	3.7495	1.9259	4.2026
C2	2.5182	2.6855	1.2863	0.4689
C3	0.1553	2.3214	3.6820	3.1892
C4	1.5163	1.8333	4.9295	4.6926

C.1) Hxy for full time cycle 3

	C1	C2	C3	C4
C1	1.0000	0	0	0
C2	0	1.0000	0	0
C3	0	0	1.0000	1.3011
C4	0	0	1.0875	1.0000
Mean Hxy	1.0000	1.0000	1.0438	1.1505
Sampling Intervals	10	10	10	10

C.2) Significance Level for full time cycle 3

	C1	C2	C3	C4
C1	3.0739	1.8645	0.3686	0.9108
C2	1.9590	3.3620	1.4121	1.1023
C3	0.7805	1.6623	3.1901	2.3902
C4	0.6126	1.7864	2.9865	2.7199

D.1) Hxy for full time cycle 4

	C1	C2	C3	C4
C1	1.0000	1.4125	2.2298	2.4301
C2	1.3991	1.0000	1.4035	1.5881
C3	2.1466	1.3903	1.0000	1.0462
C4	2.2881	1.5559	1.0349	1.0000
Mean Hxy	1.7084	1.3397	1.4171	1.5161
Sampling Intervals	20	10	10	20

D.2) Significance Level for full time cycle 4

	C1	C2	C3	C4
C1	3.4077	3.2441	2.0413	2.1554
C2	2.9505	3.5783	2.9398	2.4291
C3	2.9768	3.1715	6.2943	5.6282
C4	2.0533	2.7170	4.9513	4.9255

E.1) Hxy for full time cycle 5

	C1	C2	C3	C4
C1	1.0000	1.3795	0	0
C2	1.3791	1.0000	1.5471	1.6817
C3	2.0827	1.4760	1.0000	1.0338
C4	2.2282	1.6126	1.0594	1.0000
Mean Hxy	1.6725	1.3670	1.2022	1.2385
Sampling Intervals	20	10	10	10

E.2) Significance Level for full time cycle 5

	C1	C2	C3	C4
C1	4.1398	3.7897	1.6961	1.6341
C2	4.3974	4.4747	3.6149	3.1423
C3	3.1842	5.5412	7.1207	6.2049
C4	3.6734	4.5311	6.5329	6.4822

Figure 7-6 Predictability matrices and significance levels for consecutive full time cycles

The average predictability measures (denoted by Mean Hxy) amongst interdependent nodes and the corresponding adaptive sampling intervals generated for each of the $adapt_{sense}$ time cycles are also shown in the Figure 7-6. The average

predictability measures are computed across each column j. The node that exhibits the lowest average predictability measure exerts the maximum influence on the other nodes, represented by C_k in equation (7-15) and is assigned the lowest sampling interval equal to 10s (embedding delay value).The remaining nodes are assigned the adaptive sampling intervals calculated based on their average predictability measure using the relationship in equation (7-15).

For example in matrix A.1, nodes C_2, C_3, have the lowest average predictability measures and hence are assigned the lowest sampling interval given by embedding delay (10s). Node C_2 is assigned the lowest sampling interval by virtue of the fact that it does not influence any of the neighbouring nodes and measures data significantly different from them. Node C_3 has the lowest average predictability measure (1.0768) and is therefore assigned the lowest sampling rate. Nodes C_1, C_4 are allocated sampling intervals 30s and 20s each in proportion to their average predictability measures (2.8309*10s=30s and 2.3545*10s=20s rounded off to the nearest multiple of 10). Hence, it can be seen that at every adaptive time cycle, the nodes are assigned sampling intervals in accordance with the dynamic data changes and the resulting interdependence relationship derived between the data from different nodes.

7.4 Performance evaluation of NNASS

In this section, the performance evaluation of NNASS is carried out. In Section 7.4.1, the selection of various NNASS algorithm parameters is explained. The effect of variation of the different network parameters on the algorithm performance is further investigated in sub-section 7.4.1.2. The performance comparison against ASAP algorithm is carried out in Section 7.4.2.

The metrics that are used for NNASS performance evaluation are as follows:

1. *Sampled data reduction*: The sampled data reduction for each node is calculated as a ratio between the number of data points not sampled by the sampling algorithm and the actual data points in the dataset. The length of the Cyprus dataset is 5 hours (3600*5= 18000 samples) and Indian dataset is 8 hours (3600*8=28800 samples) with samples taken every 1s. An average sampled data

reduction is reported across all the nodes to give an estimate of the sampled data reduction in the overall network.

2. *Mean deviations*: The mean deviations represent the average of the percentage differences computed between the sampled and real data points over the different time cycles in a dataset for each node. For example, for a dataset with 18000 data points and a time scale of 900s, there are twenty time cycles. For each of the time cycles, the percentage difference between the sampled and real data is computed. An average value of the percentage differences accumulated across all the time cycles represents the mean deviations for the node. Finally, an average mean deviation value is reported across all the nodes to give an estimate of the overall mean deviations achieved across the whole network. It should be noted that the missing points in the sampled data time series are reconstructed using the "sample and hold" technique.
3. *Sampling performance*: Sampling performance is evaluated using a ratio of the average sampled data reduction and mean deviations computed over the whole network. This metric is used for evaluating the trade-off between the sensor energy savings and data accuracy. Higher values imply better sampling performance. This metric is used for comparison between the different algorithm parameter values and performance comparison against the ASAP algorithm.

In this work, the prime interest is on evaluating the trade-off between sensor energy savings obtained due to sampled data reduction and data quality provided by the sampling algorithm. Therefore, the clustering is initially performed and then the clusters are assumed to stay intact for a certain time interval, until the next clustering is performed. The messaging overhead owing to the clustering and the data aggregation has not been the focus of this analysis.

7.4.1 Parameter selection for NNASS algorithm

There are several different parameters in the NNASS algorithm that need to be appropriately selected for the pollution data. The various parameters are as follows:

1. Embedding dimension, m
2. Prediction horizon, h
3. Time scale, s

4. Embedding delay, τ
5. Number of nearest neighbours, K

Other than these, there are network parameters – the number of nodes, N and the transmission radius, R that need to be taken into account in order to understand their impact on the NNASS spatial behaviour. The selection of the default values for each of these parameters is explored in this section.

Selection of the embedding dimension: The embedding dimension is one of the parameters that capture the dynamics of the time series. The embedding dimension m for the pollution dimension is estimated using the correlation dimension method as explained in Section 4.5. According to the correlation dimension method, the saturation value of the correlation exponent is defined as the correlation dimension of the attractor, and the nearest integer above the saturation value provides the minimum number of the embedding dimensions of the phase-space required to model the dynamics of the attractor. More details can be found in Section 3.6. Based upon the correlation dimension method, an embedding dimension of 3 is fixed for the pollution data.

Selection of the prediction horizon: The prediction horizon is fixed to 1, i.e. only the next/single step forecasts are calculated to be used in the cross-prediction as explained in Section 7.2.1. The forecasts are generated using the mean value across all the K embedding vectors.

Selection of the time scale: The time scale s is important to compute the number of samples used for finding the predictability measure. It has an impact on the magnitude of the results and more data should give a better estimate of the predictability measure. The 20-minute (1200s) statistical requirement is specified by the environmental engineers [12],[13] as the time scale of interest in their studies. Hence, a 20-minute time scale is fixed in the experiments in this chapter.

These default parameter values for the embedding dimension m, the prediction horizon h and the time scale s are used in all the experiments unless stated otherwise. Given the fixed values of the embedding dimension m, the prediction horizon h and the time scale s, the embedding delay τ and the number of nearest neighbours K is varied and default values are derived experimentally for the given pollution datasets.

Selection of the embedding delay: The embedding delay is one of the most important input parameters in the NNASS algorithm. Like the embedding dimension, this parameter also captures the dynamics of the time series and determines the base sampling rate of the nodes. The embedding delay is varied from 5s to 40s. For an embedding delay of 5s in a time scale of 1200s, NNASS algorithm runs over 240 samples while for an embedding delay of 40s, there are only 30 samples for embedding representation and predictability measurement. Hence, higher embedding delay values have not been used since the number of data samples is too small to compute the predictability measure. According to the shape of the autocorrelation function (more details can be found in Section 4.5), the proper value of the embedding delay may be obtained and a common choice for the time delay is the time, at which the autocorrelation function has its first minimum, or the point at which the autocorrelation function drops exponentially to 1/e or 1/10 or 0. Based on the autocorrelation analysis of 20 min datasets as shown in Section 4.3.3, embedding delay values should be selected in the range of 20s to 30s.

Selection of the number of nearest neighbours: The number of nearest neighbours K is expected to have only a limited impact on the predictability measure since it is used only for computing the statistical averages. The number of nearest neighbours is set from 12 to 18.

Selection of the network parameters: The network topology for the Cyprus datasets is fixed to comprise of 25 nodes with transmission radius of 2m, while the network topology for the Indian datasets comprises of 50 nodes with transmission radius of 5m for these set of experiments. The selection of these network parameter values is based upon the clustering analysis carried out for EDSAS-S performance evaluation in the section 6.5. These network parameters give good clustering performance using the hierarchical agglomerative clustering and the results are comparable to the centralized affinity propagation clustering.

Spatially interpolated datasets from Cyprus and India as explained in Section 6.3 are used to evaluate the NNASS performance using the given network parameters. For each of the trials, simulations are carried out to generate different node topologies. It is followed by the creation of the clustered network architecture using the hierarchical agglomerative clustering with correlation threshold set to 0.9, so that only the most correlated nodes are clustered together. Once the clustered architecture

is created, within each of the clusters, the NNASS algorithm runs over the data of each node using the chosen algorithm parameter values and the performance metrics – sampled data reduction, mean deviations and sampling performance are recorded.

First the effect of the embedding delay and the number of nearest neighbours is evaluated using the fixed network settings. Section 7.4.1.1 gives the results for the variation of the embedding delay and the number of nearest neighbours. This is followed by the study of the effect of different network settings on the algorithm performance. Section 7.4.1.2 gives the results for the variation of the different network parameters – the number of nodes and the transmission radius.

7.4.1.1 Effect of different embedding delay and number of nearest neighbour values

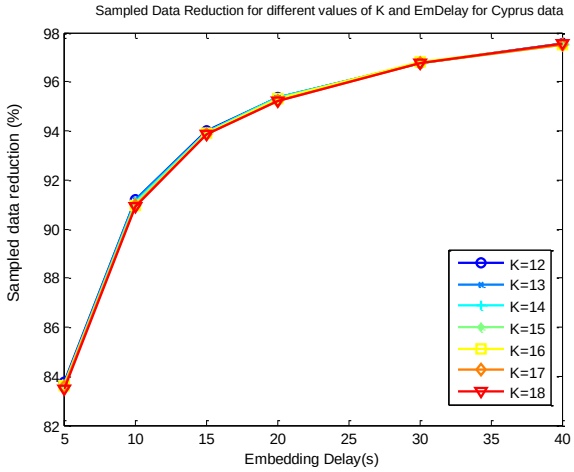
The results for the variation of the embedding delay and the number of nearest neighbours for the Cyprus dataset are shown in Figure 7-7(a)-(c). As explained in the previous Section 7.4.1, the embedding dimension is set to 3, the prediction horizon is set to 1, the time scale is set to 20min, the number of nodes is set to 25, and the transmission radius is set to 2m. With these parameter settings, the sampled data reduction, mean deviations and sampling performance metrics are evaluated for the different values of the embedding delay and the number of nearest neighbours.

It can be observed from Figure 7-7(a) that for a given number of nearest neighbours, as the embedding delay increases, the sampled data reduction first rises sharply and then become stable. The sampled data reduction increases from 84% to 94% at lower embedding delay values of 5s to 20s and increases gradually from 94% to 98% at higher embedding delay values of 30s to 40s.

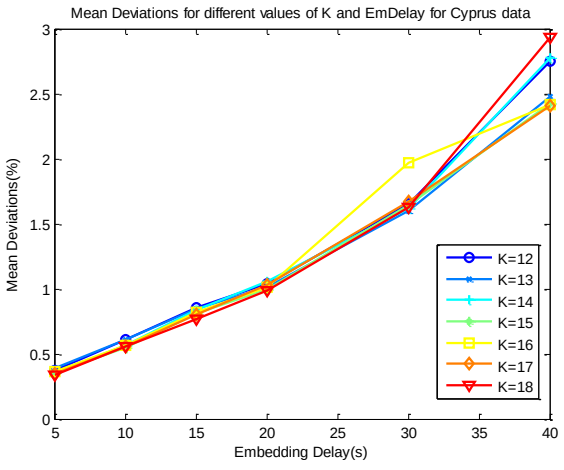
The sharp rise in sampled data reduction at lower values of embedding delay of 5s to 20s is justified because the number of actual data points per time scale decreases sharply from 240 to 60 as the embedding delay increases, which leads to higher sampled data reduction at higher embedding delay values. The adaptive sampling intervals are also proportional to the embedding delay values, so this will lead to a smaller number of sampled points as the embedding delay increases, further leading to higher sampled data reduction. Both these factors contribute towards higher sampled data reduction at higher embedding delay values. Similarly the gradual rise of the sampled data reduction as a result of changing the embedding delay from 30s

to 40s is justified because as the embedding delay increases, the total number of actual data points in a time scale falls from 40 to 30; correspondingly the sampled data reduction also change gradually.

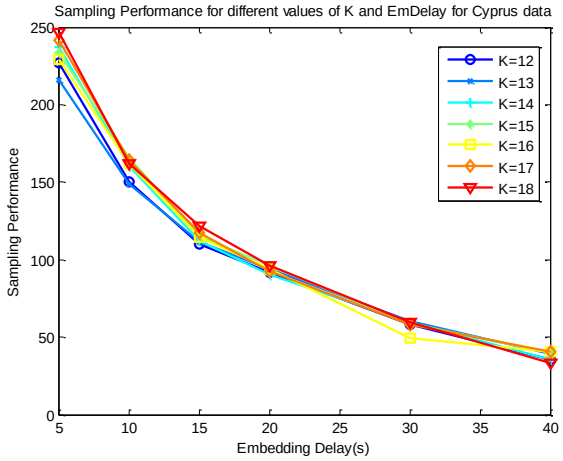
Further varying the number of nearest neighbours from 12 to 18 (shown by different colour lines), has little impact on the amount of sampled data reduction (less than 1% difference for any given embedding delay value).



(a)



(b)



(c)

Figure 7-7 (a) Sampled data reduction (b) mean deviations (c) sampling performance for Cyprus datasets for different K and embedding delay values (m=3,h=1,s=20min,N=25,R=2m)

As can be seen from the Figure 7-7(b), in the case of mean deviations for a given number of nearest neighbours, at lower embedding delay values, the mean deviations

increase gradually and stay less than 1%. As the embedding delay increases from 20s to 40s, the increase in the mean deviations is almost exponential and the mean deviations become as high as 4%. The mean deviations increase since smaller number of data points are sampled and lead to a loss in the data accuracy levels. Further, it can be seen that at embedding delay values of 5s to 20s, for different number of nearest neighbours, the mean deviations almost coincide with each other. At higher embedding delays of 30s to 40s, the mean deviations across different number of nearest neighbours oscillate.

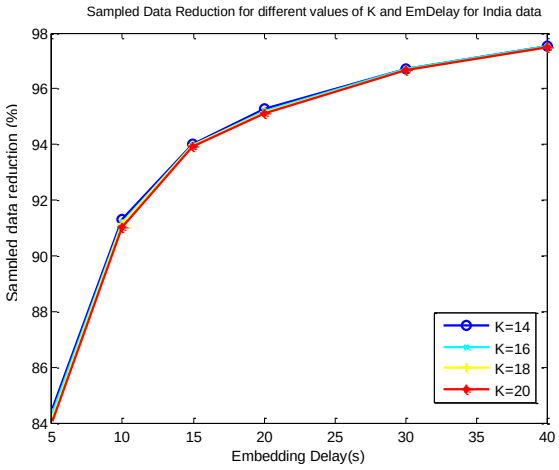
Higher sampling performance is achieved at lower embedding delay values and SP gradually decreases as the embedding delay increases. Varying the number of nearest neighbours has very little impact on the SP. In fact, the overall sampling performance for higher values of nearest neighbours is only slightly higher than the lower values of nearest neighbours as shown in Figure 7-7(c).

Next the results for the sampled data reduction, mean deviations and sampling performance for the Indian datasets are shown in Figure 7-8(a)-(c). The embedding dimension is set to 3, the prediction horizon is set to 1, the time scale is set to 1200s, the number of nodes is set to 50, and the transmission radius is set to 5m.

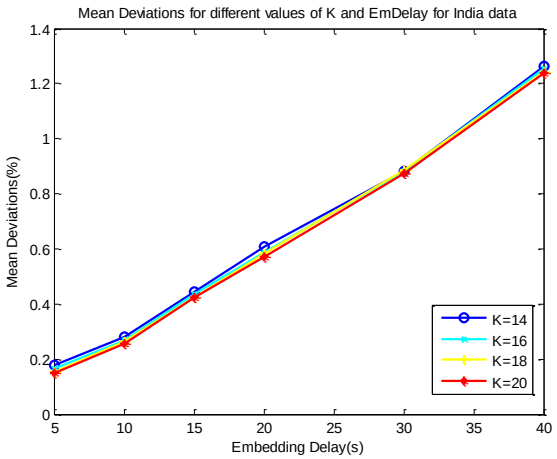
The performance results for the Indian datasets are similar in behaviour to the Cyprus datasets. It can be seen from Figure 7-8(a) that in case of lower embedding delay values, the sampled data reduction obtained for a given nearest neighbour value are much lower than in case of higher embedding delays, but the increase in sampled data reduction is sharper (84% to 94%) at lower embedding delay values of 5s to15s. At the higher embedding delays of 20s to 40s, the potential gain in sampled data reduction almost becomes constant (95% to 97%). It can also be observed in this case the sampled data reduction is almost the same for the different number of nearest neighbours.

It can be seen from Figure 7-8(b) that in case of the Indian datasets, the mean deviations obtained for a given nearest neighbour value are lower than the Cyprus datasets, and until the embedding delay value reaches 30s, the mean deviations stay below 1%, and the increase in mean deviations is almost linear in nature. These results indicate that low embedding delay values are preferable if higher sampling

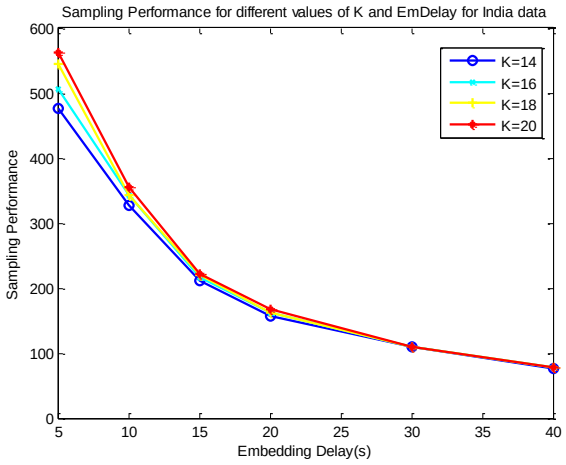
performance is desired. It can also be observed that the mean deviations are almost the same for the different number of nearest neighbours.



(a)



(b)



(c)

Figure 7-8 (a) Sampled data reduction (b) mean deviations (c) sampling performance for India datasets for different K and embedding delay values (m=3,h=1,s=20min,N=50,R=5m)

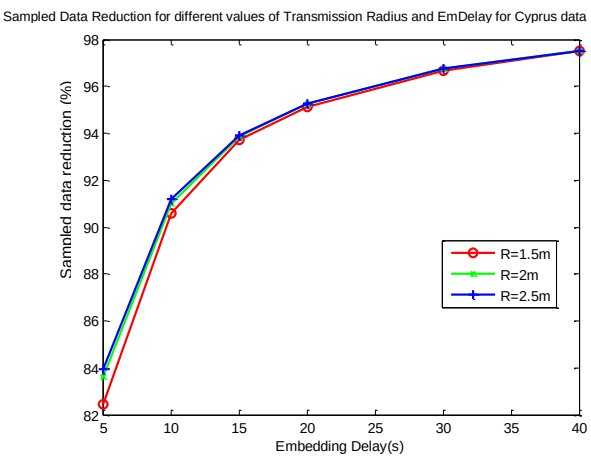
At lower embedding delay values, the sampling performance for a higher value of nearest neighbour is only slightly higher than for those with smaller value of nearest neighbour. At higher embedding delay values, the use of different values of nearest neighbours has no impact on the sampling performance.

Therefore, based upon these experiments, the embedding delay values in the range of 5s to 20s should be used in case high data accuracy requirements are

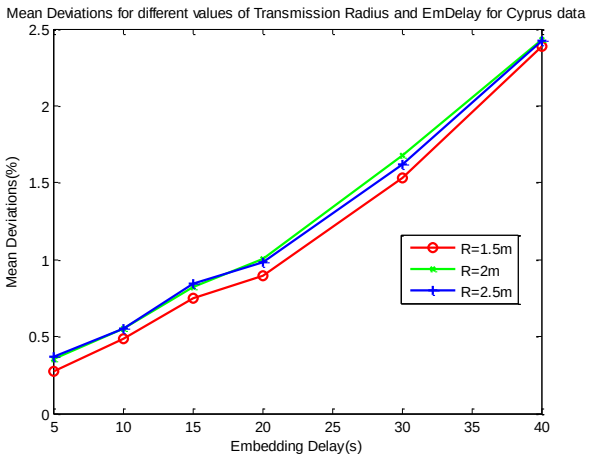
specified. For the rest of the experiments, an embedding delay value of 10s is used unless otherwise stated. The limited impact on the NNASS performance with the variation of the nearest neighbours is in line with what was expected since K is used only for computing the statistical average in the predictability measurement. K is chosen to be fixed at an intermediate value of 15. Next the impact of different network parameters – the number of nodes and the transmission radius on the performance metrics is shown.

7.4.1.2 Effect of different network parameters

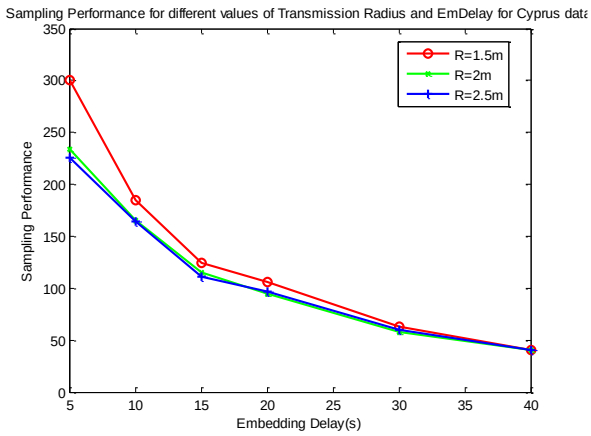
First the result of the variation of network parameters is shown for the Cyprus datasets. The effect of the transmission radius is evaluated by varying R from 1.5m to 2.5m. The number of nodes is set to 25 and the number of the nearest neighbours is set to 15. It can be seen from Figure 7-9(a) and (b) that at embedding delay values of 5s to 20s, both the sampled data reduction and the mean deviations at shorter transmission radius (1.5m) are lower in comparison to the longer transmission radius (2m and 2.5m). The reason is that more clusters are formed, but they are smaller in size due to the shorter transmission radius and therefore, more nodes will sample at the base sampling rate. At embedding delay values of 30s to 40s, the sampled data reduction and the mean deviations almost coincide for the different transmission radii. This indicates that the effect of varying the transmission radius diminishes at higher embedding delay values. This is due to the reason that at higher embedding delay values, the effect of the embedding delay masks the effect of the transmission radius.



(a)



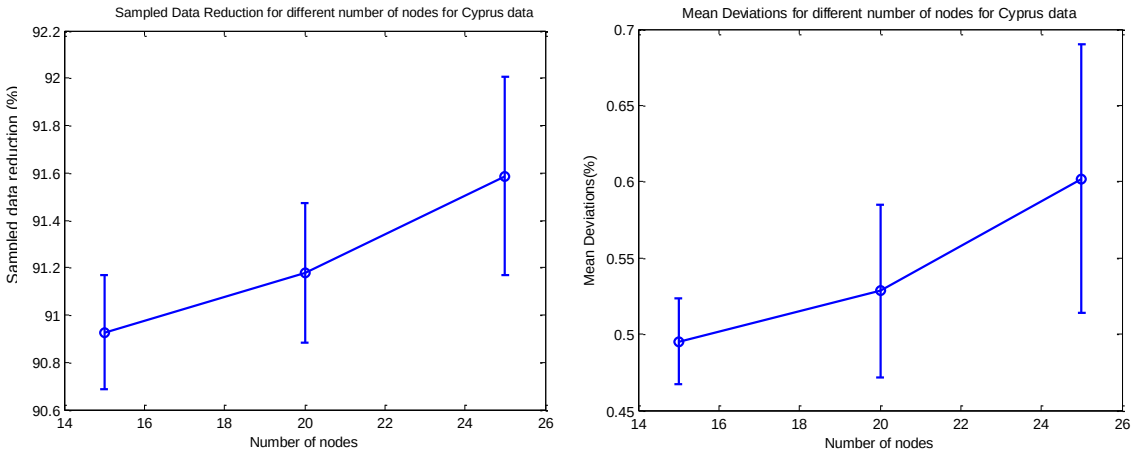
(b)



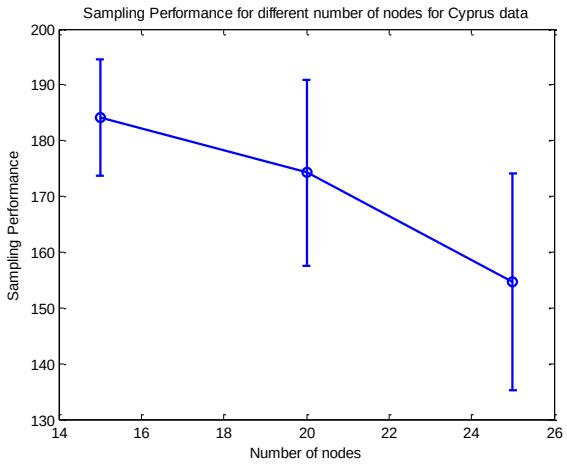
(c)

Figure 7-9 (a) Sampled data reduction (b) mean deviations (c) sampling performance for Cyprus datasets for different transmission radius and embedding delay values (m=3,h=1,s=20min,N=25,K=15)

Next the effect of number of nodes is evaluated by changing N from 15 to 25 for the Cyprus datasets in Figure 7-10(a)-(c). The effect of different topologies is evaluated in these experiments by carrying out ten different simulations. For each simulation, random node locations are generated using the spatially interpolated datasets as explained in Section 6.3 and produce different network topologies. The embedding delay is set to 10s, the transmission radius is set to 2m, and the number of nearest neighbours is set to 15.



(a) (b)



(c)

Figure 7-10 (a) Sampled data reduction (b) mean deviations (c) sampling performance for Cyprus datasets for different number of nodes (m=3,h=1,s=20min,τ=10s,K=15,R=2m)

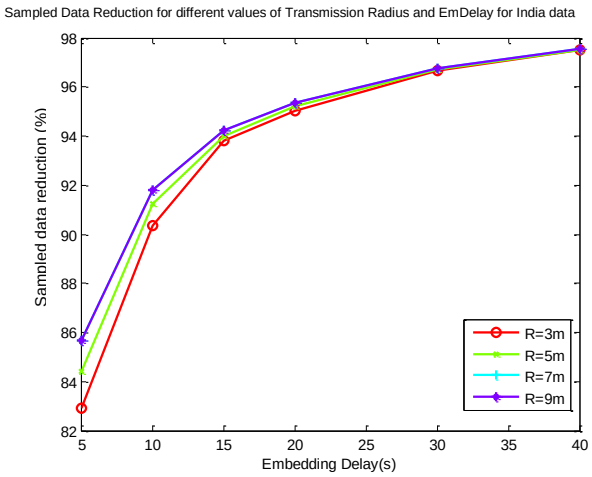
It can be observed from Figure 7-10(a) and (b) that as the number of nodes increase, the sampled data reduction and the mean deviations both increase very slightly (0.5% increase in sampled data reduction and 0.2% in mean deviations). The error bars depict the standard deviation in the results obtained across the ten different simulations. It can be seen that the difference in sampled data reduction due to the different network topologies is only 0.5% to 1% and in the case of mean deviations, the difference is even smaller from around 0.1% to 0.2%.

As the number of nodes increase, the average cluster size increases, which means there are more nodes per cluster. NNASS allocates the sampling intervals to all the cluster nodes in accordance with the predictability levels between the pairs of nodes. This leads to more cluster nodes being sampled at higher sampling intervals, consequently there is higher sampled data reduction and higher mean deviations.

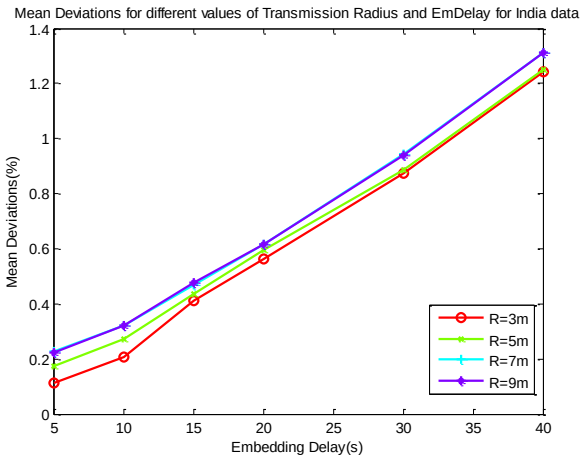
Next, the effect of different network parameters on the Indian datasets is shown. First the effect of the variation of the embedding delay and the transmission radius (3m to 9m) is shown in Figure 7-11(a)-(c). The number of nodes is set to 50 and the number of nearest neighbours is set to 15.

It can be observed that in this case, the sampled data reduction for shorter transmission radius is smaller than for longer transmission radius. There are more

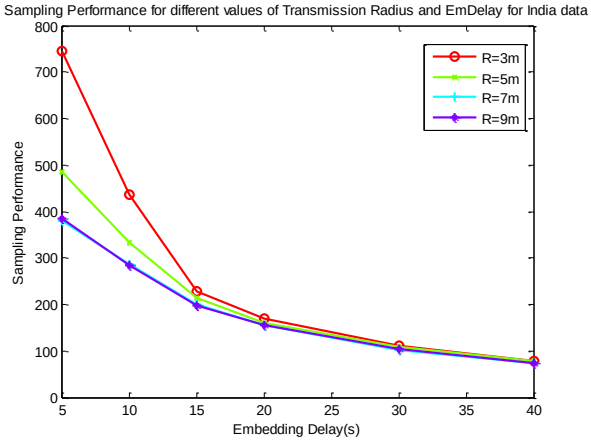
clusters formed at shorter transmission radius, which leads to more nodes being sampled at lower sampling intervals. Hence this cause lower sampled data reduction and mean deviations at shorter transmission radius.



(a)



(b)



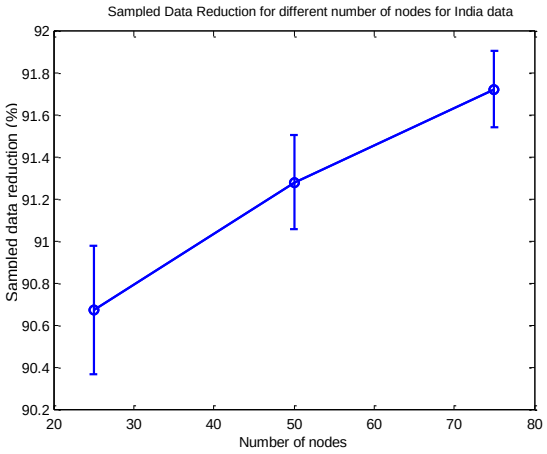
(c)

Figure 7-11 (a) Sampled data reduction (b) mean deviations (c) sampling performance for India datasets for different transmission radius and embedding delay values (m=3,h=1,s=20min,N=50,K=15)

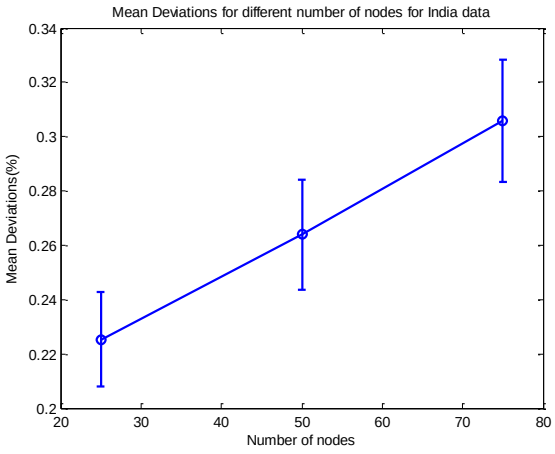
Next, the results for the variation of the number of nodes for the Indian datasets are shown in Figure 7-12(a)-(c). In this case the number of nodes varies from 25 to 75 and the transmission radius is kept fixed at 5m. The embedding delay is set to 10s and the number of nearest neighbours to 15. As the number of nodes increases, the

network node density increases and the nodes located close to each other are clustered together.

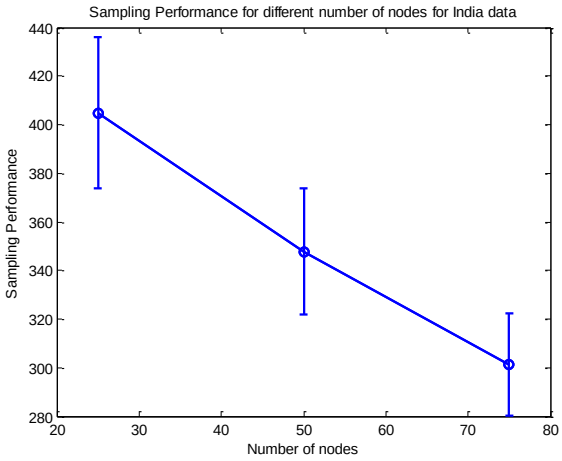
The overall sampled data reduction and the mean deviations increase as the number of nodes increase because the average cluster size increases with more nodes per cluster. So, there are more nodes per cluster that sample at higher sampling intervals resulting in higher sampled data reduction and mean deviations. The overall sampled data reduction stay within 90% to 92% with mean deviations varying from 0.2% to 0.3%.



(a)



(b)



(c)

Figure 7-12 (a) Sampled data reduction (b) mean deviations (c) sampling performance for India datasets for different number of nodes ($m=3, h=1, s=20min, \tau=10s, K=15, R=5m$)

Ten different network simulations are carried out using the spatially interpolated datasets. The error bars in Figure 7-12 show the result of the variations of the network topologies for the different number of nodes. It can be seen the bars depict higher variations in the sampled data reduction at smaller number of nodes than at larger number of nodes. The reason is that in a high network density, the variation effect caused by different topologies is smaller. When the network density is lower, different network topologies can cause more variations (a variation in sampled data reduction from 0.4% to 0.6%) across the different topologies.

These set of experiments prove that for the different network parameters, the NNASS performance behaviour can be summarized as follows:

1. As the number of nodes increases, the sampled data reduction and the mean deviations increase.
2. At smaller embedding delay values, the sampled data reduction and the corresponding mean deviations for shorter transmission radius is lower in comparison to longer transmission radius. At higher embedding delay values, the sampling performance does not vary much across the different transmission radii.

7.4.2 Performance comparison against ASAP

The ASAP [85] algorithm is chosen for comparison with NNASS because ASAP uses a clustering mechanism to incorporate the spatial correlations and every update time cycle (similar to the time scale s in NNASS), updates the sampler and non-sampler nodes based on the temporal correlations. A brief introduction to the ASAP sampling techniques was given in Section 6.2 on the related work on spatial sampling techniques. Here, more detail about the ASAP technique is provided in sub-section 7.4.2.1. Sub-section 7.4.2.2 provides the actual performance comparison results between NNASS and ASAP.

7.4.2.1 The ASAP spatial sampling algorithm

In ASAP, the first step is to use a *sensing driven clustering* algorithm to create the clusters based on the hop distances and the spatial data correlations. The authors of ASAP have proposed to perform the clustering after a fixed time interval denoted as the clustering period τ_c. For instance in environmental monitoring applications, different times of a day may result in different clusters. Thus, the clustering period

τ_c should be adjusted accordingly to enable continued refinement of the clustering structure in response to the different sensing patterns resulting from the environmental changes. This clustering process achieves the same purpose as the hierarchical agglomerative clustering performed in NNASS. The clustering process yields clusters or groups of nodes with similar data readings. For the sake of comparison of both the ASAP and NNASS algorithms, the hierarchical clustering is performed over the whole network and the obtained clustered architecture is assumed to stay intact for the whole sampling duration. However, more details about the ASAP's sensing driven clustering can be found in Section 6.2.2.

The second step is to create the sub-clusters for each of the node clusters. The goal of further dividing the node clusters into sub-clusters is to facilitate the selection of the nodes to serve as the samplers and the generation of the probabilistic models for value prediction of the non-sampler nodes. This is achieved by the *correlation-based sampler selection* and *model derivation algorithm* that is executed periodically at every τ_u seconds. τ_u is called the schedule update period. The cluster head node carries out the sampler selection and model derivation task locally in three steps: (1) The cluster head node uses infrequently sampled readings from all the nodes within its cluster to capture the spatial and temporal correlations in the sensor readings and calculate the sub-clusters, so that the nodes whose sensor readings are highly correlated are put into the same sub-clusters. (2) The sub-clusters are used to select a set of sampler nodes such that there is at least one sampler node selected from each sub-cluster. This selection of samplers forms the sampling schedule for the cluster. A system-wide parameter $\sigma \in (0,1]$ is defined as the average fraction of nodes that should be used as samplers. σ is called the sampling fraction. Once the sampler nodes are determined, only these nodes report sensor readings to the base node and the values of the non-sampler nodes will be predicted at the sink (or the base node) using probabilistic Multi-Variate Normal (MVN) models [85] that are constructed in an in-network manner for each sub-cluster and reported to the base node. (3) A probabilistic model is constructed and reported for each sub-cluster within the network. A system-supplied parameter β is introduced, which defines the average size of the sub-clusters. β is called the sub-cluster granularity and its setting influences the size and number of the sub-clusters used in the network.

The third step is to collect the sensed values from the network and to perform the prediction after the sensor readings are received. This is achieved by the *adaptive data collection* and *model-based prediction* algorithm. The adaptive data collection component works in two steps: (1) Each sampler node senses a value every τ_d seconds, called the desired sampling period. τ_d sets the temporal resolution of the data collection. (2) In order to empower ASAP with self-adaptation, periodically, but infrequently ASAP collects (at the cluster heads) sensor readings from all nodes within a cluster at every τ_f seconds (τ_f is a multiple of τ_d). τ_f is a system-supplied parameter, called the forced sampling period. These readings are collected and used by the cluster head nodes, aiming at incorporating the newly established correlations among the sensor readings into the decision making process of the correlation-based sampler selection and model derivation algorithm. The model-based prediction component is responsible for estimating the values of the non-sampler nodes within each sub-cluster, using the readings of the sampler nodes and the parameters of the probabilistic models constructed for each sub-cluster. Figure 7-13 shows the sampling strategy of the ASAP algorithm.

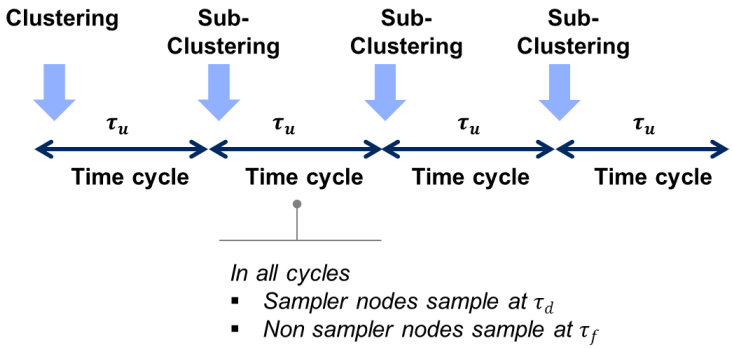


Figure 7-13 Sampling strategy of ASAP

In summary, within each of the clusters, a number of sub-clusters depending on the parameter β (sub-cluster granularity) are formed. Then within each sub-cluster, a certain fraction of nodes depending on the parameter σ (sampling fraction) are designated as sampler nodes and the remaining nodes are designated as non-sampler nodes. The sampler nodes sample data at a desired sampling period τ_d and the non-sampler nodes sample data at a forced sampling period τ_f for a fixed time interval denoted by τ_u. Within each cluster, at the end of each τ_u schedule update period, the sub-clusters are formed again depending on the data correlations and the newly

designated sampler and non-sampler nodes are sampled using the desired and the forced sampling frequency respectively.

NNASS has an analogy to ASAP since the clustering used is based on spatial correlations and once the clusters are formed, all the nodes sample for a fixed time interval (time scale s in NNASS and τ_u in ASAP). At the end of this time interval, while in NNASS, only the best predictor node samples at the lowest sampling interval (embedding delay τ), the other nodes sample at adapted sampling intervals depending on the predictability measure derived from the data. Depending on the actual data dynamics prevailing in the environment, NNASS will not only choose the most appropriate node to be sampled at the base sampling interval, but it will also assign adaptive sampling intervals to the other cluster nodes.

ASAP also adapts the sampler and non-sampler nodes at every update cycle through the sub-clustering mechanism taking the data correlations into account. However, ASAP does not use the underlying data characteristics to assign the desired and forced sampling intervals. The sampling intervals are fixed for both the sampler and non-sampler nodes at every update interval. Moreover, ASAP does not provide any application specific insight for assigning the desired and forced sampling intervals and they need to be chosen empirically.

Hence, NNASS is more adaptive than ASAP in terms of the assignments of the adaptive sampling intervals and also by sampling at the appropriate embedding delay (determined specifically for the given data at hand), NNASS captures the data dynamics more accurately. Moreover, there is an obvious additional messaging overhead in ASAP for forming sub clusters after every update cycle τ_u, but as mentioned earlier, in the current evaluation the focus is upon the sampling performance in terms of the trade-off between the sampled data reduction and the mean deviations rather than the clustering performance overheads. Therefore, the sampled data reduction and data accuracy metrics are used for a performance comparison between the two algorithms.

7.4.2.2 Performance comparisons between NNASS and ASAP

In order to carry out an unbiased comparison, the ASAP parameters are chosen in accordance with the NNASS, so that the performance given by both the algorithms is comparable. The ASAP parameters are set as follows: the update time cycle τ_u is set

to 1200s. The value of τ_u in ASAP is equal to the time scale, s used in NNASS. This means every 1200s, sub-clusters are formed again in the ASAP implementation to decide the new sampler and non-sampler nodes. The desired sampling frequency τ_d for the sampler nodes follows the embedding delay value, which varies from 5s to 40s. τ_d is set to the same value as the base sampling rate, τ in NNASS. The forced sampling frequency τ_f for non-sampler nodes is fixed to 2*τ_d, 5*τ_d and 10*τ_d. Three different multiples (2, 5 and 10) are used in order to evaluate the ASAP performance for different ratios of $\frac{\tau_f}{\tau_d}$. A multiple of 2 will give best ASAP performance for the mean deviations, while multiples of 5 and 10 imply that the forced samples are more apart in time and consequently, they will lead to higher mean deviations. Lower values of $\frac{\tau_f}{\tau_d}$ also mean there will be more forced samples, so the sampled data reduction will be lower in compassion to higher values of $\frac{\tau_f}{\tau_d}$.

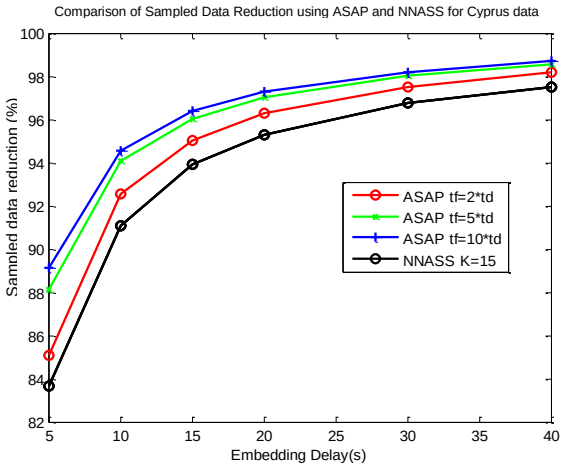
The sampling fraction parameter σ is set to 0.25, which means only 25% of the nodes in a sub cluster will be designated as the sampler nodes. If σ is set to a higher value, it will lead to more nodes being chosen as sampler nodes and therefore lead to lower data reduction. The authors in [85] have assumed a default average cluster size of 30 and have chosen the value of sub-cluster granularity parameter β to 10 in the simulations; this means that the number of sub clusters is set to 3. Based on a similar insight to get 3 sub-clusters for the spatially interpolated pollution datasets, the sub-cluster granularity parameter β is set to 3 in these performance comparison experiments. Now the results for sampled data reduction, mean deviations and sampling performance for both the algorithms for the Cyprus datasets using the parameter settings explained above are shown in Figure 7-14(a)-(c).

It can be seen from the graph in Figure 7-14(a) that the sampled data reduction for NNASS for any given value of embedding delay is always lower than the ones given by ASAP for a corresponding value of τ_d. The sampled data reduction given by NNASS is close to those given by ASAP for $\frac{\tau_f}{\tau_d} = 2$, while for $\frac{\tau_f}{\tau_d} = 5$ or $\frac{\tau_f}{\tau_d} = 10$ the sampled data reduction given by ASAP is higher by an order of 2% to 6% in comparison to NNASS. This is because ASAP uses higher values of τ_f and it samples less than NNASS. The sampling intervals are not adaptive in case of ASAP, while in

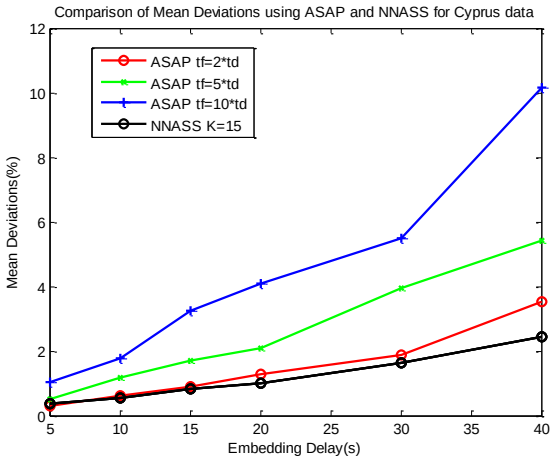
the case of NNASS, the sampling intervals are assigned based on the predictability measure derived from the data interdependencies.

In the case of the mean deviations as shown in Figure 7-14(b), NNASS has the lowest mean deviations in comparison to ASAP for all the values of $\frac{\tau_f}{\tau_d}$. The reason for higher mean deviations of ASAP is again because less data points are sampled and the higher forced sampling intervals leads to higher mean deviations as it can be seen in the case for $\frac{\tau_f}{\tau_d} = 5$ or $\frac{\tau_f}{\tau_d} = 10$ when mean deviations are as high as 6% to10%. Overall this leads to a higher sampling performance by NNASS than ASAP for all the embedding delay values other than the lowest embedding delay value of 5s, when ASAP gives higher performance due to the higher data reduction values.

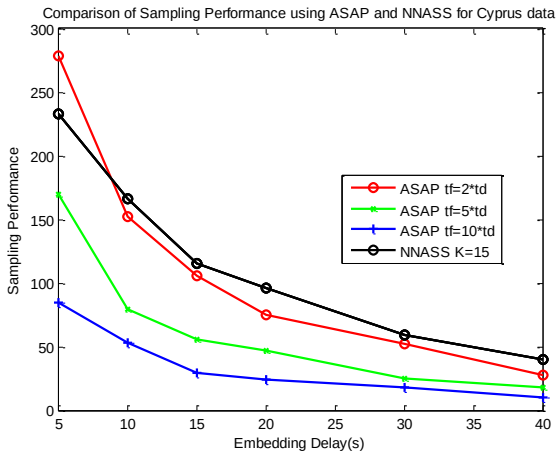
It is important to estimate that the difference in sampling performance for the two algorithms is statistically significant or not in order to give conclusive results about the algorithm performance. A *paired sample t-test* [86] can be used to determine whether there is a significant difference between the sampling performances obtained from the two algorithms. The t-test gives a decision for the null hypothesis that the difference in sampling performance comes from a normal distribution with mean equal to zero and unknown variance. If the p-value is less than 0.05, it indicates that the null hypothesis can be rejected and the difference in sampling performance is statistically significant. The t-test carried out for the sampling performance given by ASAP and NNASS yields p-values as shown in Table 7-1 for the Cyprus and Indian datasets.



(a)



(b)



(c)

Figure 7-14 Comparison of ASAP and NNASS performance for Cyprus datasets

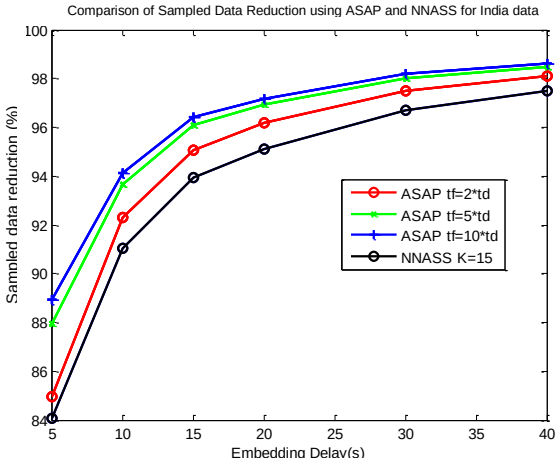
The t-test for the Cyprus datasets indicates that the difference in the sampling performance of NNASS and ASAP τ_f=2*τ_d is statistically not significant, while for ASAP τ_f=5*τ_d and ASAP τ_f=10*τ_d , the difference in the sampling performance is statistically significant. So, the performance of NNASS is comparable to ASAP for τ_f=2*τ_d and better than ASAP when τ_f=5*τ_d or τ_f=10*τ_d.

Table 7-1 Statistical significance for difference of the sampling performance between ASAP and NNASS

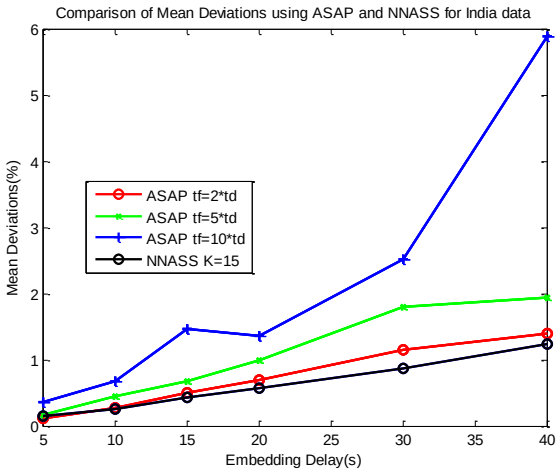
	p-value for Cyprus datasets	**p-value for India datasets**
NNASS/ASAP τ_f=2*τ_d	0.7770	0.8357
NNASS/ASAP τ_f=5*τ_d	0.0025	0.0086
NNASS/ASAP τ_f=10*τ_d	0.0063	0.0130

Next the results for sampled data reduction, mean deviations and sampling performance for both the algorithms for the Indian datasets are shown in Figure 7-15(a)-(c). It can be observed that in the case of the Indian pollution datasets, the sampled data reduction given by NNASS is also the lowest in comparison to ASAP, and the corresponding mean deviations are the lowest too. When the sampling performance for both the algorithms is compared, it can be seen that NNASS gives higher sampling performance in comparison to ASAP for $\frac{\tau_f}{\tau_d} = 2$ at all embedding delay values other than the lowest embedding delay value of 5s.

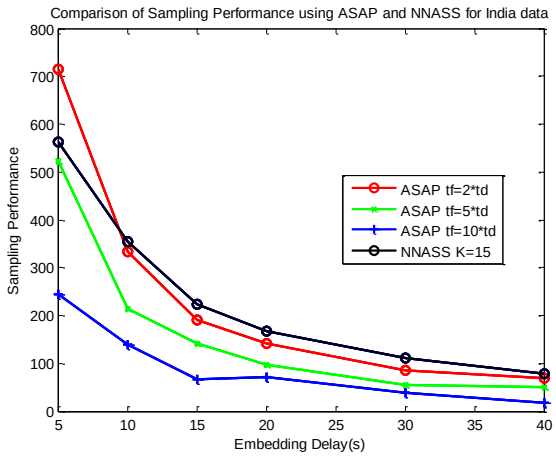
The t-test for the Indian datasets as shown in Table 7-1 indicates that the difference in the sampling performance of NNASS and ASAP τ_f=2*τ_d is statistically not significant, while for ASAP τ_f=5*τ_d and ASAP τ_f=10*τ_d , the difference in the sampling performance is statistically significant. So, the performance of NNASS is comparable to ASAP for τ_f=2*τ_d and better than ASAP when τ_f=5*τ_d or τ_f=10*τ_d.



(a)



(b)



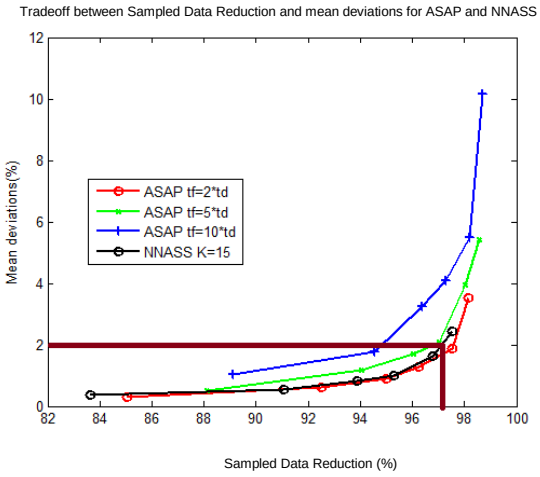
(c)

Figure 7-15 Comparison of ASAP and NNASS performance for India datasets

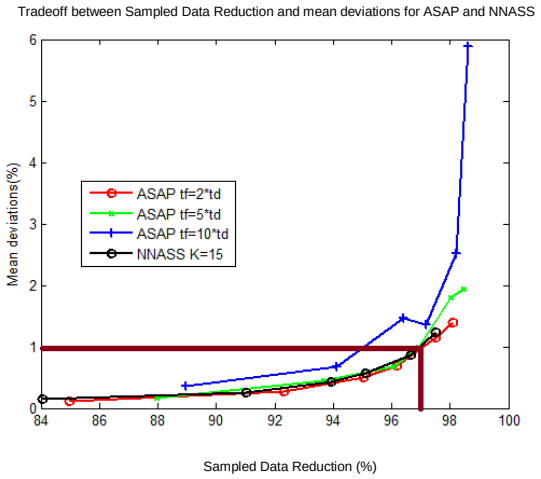
While the above analysis compares the sampling performance of the two algorithms for any given embedding delay/τ_d value, it is expected that in real life users might be more interested in finding out the appropriate input parameters to run any chosen algorithm (ASAP or NNASS) given their desired data reduction target or maximum mean deviations tolerance. In order to facilitate such a decision the

following graphs are presented that show the trade-off between the sampled data reduction and the mean deviations for the two algorithms for different embedding delay values.

Figure 7-16(a)-(b) shows the trade-off between sampled data reduction and mean deviations for both the NNASS and ASAP algorithms for both pollution datasets. The different points along the graph lines depict the different embedding delay values 5s, 10s, 15s, 20s, 30s, 40s. For a given data accuracy requirement by the user, these graphs show how much sampled data reduction can be achieved and also provide an insight into the default embedding delay parameter values.



(a)



(b)

Figure 7-16 Trade-off between the sampled data reduction and data accuracy for (a) Cyprus (b) India datasets

Figure 7-16(a) leads to the conclusion that for NNASS (black line) in the case of the Cyprus datasets, a sampled data reduction of 97% can be achieved for a data accuracy requirement of 2%. The embedding delay needs to be set to 30s. In order to achieve lower data accuracy levels (i.e. higher mean deviations), more sampled data reduction can be achieved by using embedding delay values higher than 30s. At lower embedding delay values of 15s and 20s, sampled data reduction of 94% to 95% can be obtained for less than 1% mean deviations. In the case of the ASAP, for data accuracy levels of 2%, sampled data reduction is dependent on the $\frac{\tau_f}{\tau_d}$ ratio and can vary from 94% to less than 98%.

Similarly in case of the Indian datasets as shown in Figure 7-16(b), an embedding delay of 30s can yield mean deviations of 1% and sampled data reduction of 97%. For any further sampled data reduction at lower data accuracy levels, the embedding delay needs to be set to a higher value. In the case of ASAP, sampled data reduction of 95% to 97% can be achieved for the different values of the $\frac{\tau_f}{\tau_d}$ ratio.

7.5 Limitations of the NNASS algorithm

One of the problems with the NN based predictability measure is the high computational requirements. As the average cluster size becomes larger, the NN measure needs to be computed for a greater number of pairs of nodes and therefore, takes a longer execution time. The simulations carried out in this chapter are run over small sized networks in which the average cluster size is limited. In order to take care of the NNASS execution time and computational power requirements, either the spatial clustering mechanism should have an inbuilt mechanism to limit the cluster size or a dedicated higher resource node should be employed for computing the NN based predictability measure.

Another limitation of this method is the requirement of reliable and close to ground truth data to compute the predictability measure accurately. In this work, the emphasis is to gather accurate data and the sensor devices had been adequately calibrated [26]. Also, the datasets are analysed and cleaned offline before applying the NN based predictability measurement. Hence in any such data intensive task, data reliability and trustworthiness should be taken into account.

7.6 Chapter conclusions

This chapter has presented the details of a novel spatial sampling algorithm called *Nearest Neighbours based Adaptive Spatial Sampling* (NNASS). Based on the existence of non-linear properties in pollution datasets, it has been proposed to use a nearest neighbour based predictability measure for the design of the spatial sampling algorithm. The predictability measure computation uses a time delay embedding to capture the non-linear dynamics between a given pairs of nodes. The sensor nodes that exhibit higher predictability levels are assigned lower sampling intervals, whereas the remaining nodes are assigned higher sampling intervals given that the lower predictable nodes can be adequately reconstructed from the best predictable nodes.

The algorithm has been evaluated using the spatially interpolated datasets from the Cyprus and India pollution trials. The NNASS performance has been evaluated over different algorithm parameters. Lower embedding delay values result in good sampling performance. Performance comparison is carried out against another spatial sampling algorithm called ASAP, and NNASS has either comparable or better sampling performance than ASAP depending on the choice of ASAP parameters. In the case of the NNASS, not only the sampling rate can be adapted in accordance with the data dynamics, but also the node that is the best predictor is sampled at the lowest sampling interval. In the case of the ASAP algorithm, the sampling intervals are fixed and only the sampler nodes are changed by means of the in-built sub clustering mechanism.

Based on the performance comparison study, a trade-off analysis between the sampled data reduction and mean deviations has been carried out. It provided an insight into the amount of sampled data reduction that can be achieved and default parameter values to be used for a certain data accuracy requirements provided by the user.

Chapter 8

Thesis summary and future work

8.1 Summary

Vehicular air pollution in urban street canyons is a grave problem being faced by populations around the world and wireless sensor networks (WSN) can play an integral part in its better understanding and solution. WSNs can enable the collection of fine-grained pollution data that can be used to carry out pollution studies at micro-environmental levels by the environmental scientists leading to a better understanding of the phenomena and ultimately in the design of solutions to help address the problem or reduce its harms.

This research work was carried out under the aegis of the IUTAC and formed part of the project theme - Pervasive sensing environment. As part of the project first low cost pollution monitors were developed, and deployed at two different locations – Nicosia, Cyprus and Hyderabad, India for pollution trials, and then the collected pollution datasets were used for the design and performance evaluation of novel algorithms for sensor energy management.

The main research objective in this thesis was the design, the implementation and the evaluation of novel efficient adaptive sampling techniques - in both temporal and spatial domains- for pollution sensor nodes in a WSN. Efficient adaptive sampling mechanisms are at the heart of designing systems that can save the precious sensor energy while still not losing data accuracy. Thus enabling the collection of accurate fine grained data, while ensuring larger battery life. A better understanding of the existing data characteristics of the pollution datasets (for example, non-linear dynamics) can be exploited to sample only the most relevant data points, but still provide valuable statistics to environmental scientists.

In this thesis, the important characteristics of pollution dispersion in urban street canyons were studied so as to understand the relevance of WSNs for pollution monitoring. The characteristics of the pollution datasets were investigated by using

different techniques from time series analysis, multi-fractal analysis and non-linear dynamical systems. The data analysis provided an insight into the factors affecting the pollution dynamics and guiding principles for the sampling algorithm design. The thesis presented the design and evaluation of a novel temporal adaptive sampling algorithm, EDSAS that exploits the presence of locally linear trends and slowly decaying auto correlations. It was followed by the application of the temporal sampling algorithm, EDSAS to the spatial domain. Two spatial clustering algorithms that cluster nodes based on the spatial data correlations (the hierarchical agglomerative clustering and the affinity propagation) were explored as part of a novel spatial sampling algorithm design. Also, the design of a novel spatial sampling algorithm called NNASS was proposed for pollution data. NNASS captures the non-linear dynamics, which is another interesting characteristic present in the pollution datasets.

In summary, the major research contributions of this thesis are as follows:

1. *Data analysis of fine-grained pollution datasets*

Normally environmental scientists do not have access to fine grained pollution datasets and data sets used in atmospheric sciences literature is sampled at higher sampling intervals. In the current work, due to the availability of custom made CO monitors and on board memory storage, data has been sampled at 1Hz. The datasets hence obtained have been studied and their important features and characteristics have been investigated using time series analysis.

Exploratory data analysis technique has been used to find the existence of very slowly decaying autocorrelations and locally linear trends in the pollution datasets. Also, these datasets exhibit lognormal distributions. Further, the existence of long range dependence and self-similarity has been proven by using multi-fractal de-trended fluctuation analysis. The pollution datasets have shown to possess Hurst exponent greater than 0.9 and exhibit multi-fractal characteristics too. Non-linear dynamics are present and time delay embedding parameters, i.e. embedding dimension and delay, has been investigated too. Pollution datasets used in the study exhibit an embedding dimension of three and low embedding delay values of the order of 20s to 40s. This data analysis gave a better understanding of the underlying

pollution data characteristics and how they could be exploited to design better sampling algorithms.

2. *Proposition, design and evaluation of an efficient temporal sampling algorithm called EDSAS*

Due to the presence of slowly decaying temporal correlations and locally linear trends in the pollution datasets, time series forecasting can be used to predict the future sample points. If the forecasting error does not exceed a given error threshold, the sampling can be avoided leading to sampled data reduction. In this work, a variant of time series forecasting based upon the exponential double smoothing that works for irregularly sampled time series and takes into account both the data level and trend values has been proposed to be used for data forecasting. The main advantages of the proposed novel temporal sampling technique called, *Exponential Double Smoothing based Adaptive Sampling* (EDSAS), are its design simplicity and low resource requirements. It does not incur high computational and communication overheads.

EDSAS has been run across the Cyprus and the India datasets and found to give good performance in terms of sampled data reduction (50% to 70% for the Cyprus datasets and 10% to 50% for the Indian datasets) for a given loss in data accuracy (10%). EDSAS has also been compared against e-Sense, which is a random walk based stochastic sampling scheduling algorithm. It has been found that EDSAS gives better sampling performance than e-Sense for the various pollution datasets. The e-Sense algorithm requires offline model training and updates, while EDSAS is a real-time method requiring no training phase. EDSAS can adapt the sampling rate of the sensor node in accordance with the current level and existing trend in the time series. The algorithm parameters need to be appropriately set according to the dynamics of the pollution prevailing at a particular location and the user defined requirements for data accuracy/energy savings. The performance across other datasets like temperature and humidity has also been evaluated and it proves that EDSAS can work well across different datasets.

The main insight that can be drawn from this work is that simple time series forecasting methods perform better than complicated modelling based adaptive sampling solutions for WSNs. A model based adaptive sampling technique requires

the model training and regular model updates; on the contrary, EDSAS uses a real-time recursive technique without additional overhead of model construction and maintenance. The implementation on real sensor nodes is feasible for this technique and can be carried out as a part of the future work.

3. Extension of EDSAS to the spatial domain

Most of the sampling techniques in the research literature are custom designed to operate either for the temporal or spatial data. Given the spatial correlations exhibited by the pollution datasets, as a part of the work carried out in this thesis, EDSAS has been extended to the spatial domain as well and termed as EDSAS-S. The neighbouring nodes can be clustered based on the data correlations and only a fraction of the nodes need to sample data using EDSAS and the remaining nodes can be put to sleep. The basic premise in the application of this approach is that the overhead caused by the additional clustering step does not exceed the benefits obtained from the savings in the sampling message communication. Hence choosing the right clustering technique is of utmost importance. In this thesis, a *hierarchical agglomerative clustering based on data correlations* has been used for spatial clustering. Another recently proposed *affinity propagation* clustering has also been studied and investigated. The hierarchical agglomerative clustering uses a simple bottom up mechanism to merge the nodes with similar readings in a distributed manner. The affinity propagation gives the best set of exemplars using message passing between pair of nodes. Though the well-chosen exemplars provided by the affinity propagation can serve the purpose of finding the representative nodes well and provides a benchmark with respect to the clustering performance; this clustering approach has high messaging overhead due to the centralized execution and takes a very long settling time to converge to the final set of clusters. On the other hand, the hierarchical agglomerative clustering gives a good clustering performance with low messaging overhead within short span of time.

The clustering performance of both the clustering mechanisms has been compared and studied using the spatially interpolated pollution datasets and also the impact on EDSAS-S performance using these clustering approaches has been studied. Based on the study, the hierarchical agglomerative clustering has been chosen in this work since its gives performance comparable to that obtained from the AP clustering. Finally, EDSAS-S performance has been evaluated using different algorithm

parameters like time scale and sampling node fraction and it has been found that the spatial sampling yields 15% higher data reduction for India datasets, and 25% higher data reduction for the Cyprus datasets, both with smaller than 10% loss in data accuracy. The savings in sampling message overhead is as high as 20% to 40% with the clustering message overhead mostly being less than 5% of the total message overhead.

4. *Proposition, design and evaluation of a novel spatial sampling algorithm called NNASS*

The extension of EDSAS into the spatial domain takes into account the correlations between the data from the adjoining sensor nodes. However, the pollution datasets possess other interesting properties like the long range dependence and the self-similarity. These properties can be exploited to give data metrics that can be used in the design of alternative spatial sampling algorithms. In this thesis, a nearest neighbour approach using the phase space representation has been used to compute the predictability measures and used for the design of a novel spatial sampling algorithm called *Nearest Neighbour based Adaptive Spatial Sampling* (NNASS). Predictability measures based on cross predictions between two time series can be computed to give an idea about how much one time series influences another one. The computation of the predictability measure takes into account the non-linear dynamics present in the pollution data. The node with higher values of predictability measure can be sampled using lower sampling intervals in comparison to the nodes with lower predictability measures.

NNASS has been evaluated using the spatially interpolated datasets using different algorithm parameters. It has also been compared against another spatial sampling algorithm called ASAP and found to give either comparable or better sampling performance depending on the choice of ASAP parameters. It has been proven that sampled data reduction of an order of 95% can be achieved for low mean deviations of 1% to 2% for different pollution datasets at sampling interval of 30s. This gives an insight into the sampling intervals for the pollution monitors that need to be selected by the environmental engineers to get a desirable data accuracy performance.

8.2 Future work

A few research directions are being specified here to carry out the future work on the temporal and spatial sampling techniques proposed in this thesis.

8.2.1 Future work related to temporal sampling

There exist numerous avenues for future work that can be done in temporal sampling and are briefly summarised as follows:

1. There are certain limitations in the existing temporal work that can be addressed as a part of the future work. For instance, in the current work, only trend has been taken into account for generating the time series forecasts using exponential double smoothing. Seasonal effects can also be taken into account and triple exponential smoothing or the Holt winter's method [32]can be exploited to derive better forecasts. Study of seasonality effects need longer datasets to be collected and analysed. Next, the EDSAS parameters have been kept the same across all the nodes in the study carried out in this thesis. The parameter values can be made adaptive and the effects on the algorithm performance can be studied further. Infact, more pollution case studies at different locations can be carried out to see the effect on EDSAS performance with varying traffic and meteorological conditions. Also, the sample and hold strategy has been used for the reconstruction of time series, but there are alternative re-construction methods [88],[89] specifically designed for irregularly sampled time series (in the time series analysis domain) that can be applied. The performance of the alternative reconstruction techniques can be evaluated and compared against the technique proposed in this work.

2. There are several new directions in which the temporal work can be extended. For instance, context based sampling adaptation. The pollution levels depend on the wind speeds prevailing in the ambient environment. Depending on the wind speed levels, the sampling rate of the sensor nodes can be adapted. Wind speed adaptation can be incorporated instead of EWMA based adaptation in the EDSAS design. Generally wind speed readings are measured using an external anemometer. It will be worth an effort to incorporate wind flow sensors in the pollution monitors and study the sensor energy costs. Another potential area of

research is to explore autoregressive fractionally integrated moving average (ARFIMA) [90],[91] models. ARFIMA models are time series models that generalise ARIMA models by allowing non-integer values of the differencing parameter. These models are useful in modelling time series with long memory and therefore can be used for modelling pollution data. ARFIMA models are expected to perform better in comparison to the ARIMA models for pollution data and can be tested against the EDS forecasting.

8.2.2 Future work related to spatial sampling

The following different areas can be further researched for spatial sampling:

1. As stated earlier, one of the main limitations of the proposed spatial sampling technique is its high computational requirements and communication overheads that were not accounted for in this research work. The feasibility of this technique needs to be proven by means of a real implementation as part of the future work.
   Also, the performance study in this thesis has been carried out using fixed time scale as stipulated by the environmental engineers, but the time scale can be made adaptive according to the existing pollution or traffic levels in the environment. Next, the TriscatteredInterp [67] has been used for spatial data interpolation in the current work. However, more advanced geo-statistical techniques [94] can be used for spatial interpolation purposes and used to carry out large scale experiments. Geo-statistical estimation is a two stage process:
   a. Studying the gathered data to establish the predictability of values from place to place in the study area; this study results in a graph known as a semi-variogram that models the difference between a value at one location and the value at another location according to the distance and direction between them.
   b. Estimating the values for the locations that have not been sampled. This process is known as *kriging*. The basic technique ‘ordinary kriging’ uses a weighted average of neighbouring samples to estimate the unknown value at a given location. Weights are optimized using the semi-variogram model, the location of the samples and all the relevant inter-relationships between known and unknown values. This technique also provides a standard error that may be used to quantify the confidence levels.

In the current thesis, the spatial reconstruction for the non-sampler nodes is based upon the linear regression relationships between the different nodes. Geo-statistical interpolation techniques can be used to reconstruct the time series for the non-sampled nodes and estimate the data accuracy levels.

2. There are several new directions in which the spatial work can be extended. For instance, exploring transfer entropy. Transfer entropy (TE) [92],[93] is an alternative measure of understanding the relationships based on information theory and it can be used for spatial sampling in a manner similar to the NN based predictability measure used in this work. TE measures the amount of information transfer between two random processes. Transfer entropy from a process X to another process Y is the amount of uncertainty reduced in the future values of Y by knowing the past values of X given the past values of Y. Alternatively, the NN technique can be studied for other application domains, for instance in order to determine the sampling requirements for healthcare sensor data that possess non-linear dynamics.

8.3 Concluding remarks

This thesis focused on the problem of senor energy management and proposed improvements in terms of reducing the number of samples, while not losing the data fidelity for pollution monitoring applications. EDSAS and NNASS, two adaptive sampling algorithms were proposed both in the temporal and spatial domains respectively and compared against alternative approaches proposed in the research literature. Both these algorithms are based on the intrinsic data characteristics of pollution data and they have shown good sampling performance across real fine grained pollution datasets.

With the emergence of data focused applications in WSNs, such data centric algorithms are desirable, so as to bridge the gap between the WSNs and other disciplines in the future. There is a lot of research scope in this area and these techniques can be further adapted to work across different application domains. Also, these data centric adaptive sampling techniques can be further integrated with data compression techniques to form a comprehensive energy saving framework for the wireless pollution sensor networks.

Bibliography

[1] A. Mainwaring, *et al.*, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, 2002, pp. 88-97.

[2] K. Martinez, *et al.*, "Glacsweb: a sensor network for hostile environments," in *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, 2004, pp. 81-87.

[3] L. Yu, *et al.*, "Real-time forest fire detection with wireless sensor networks," in *Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on*, 2005, pp. 1214-1217.

[4] J. Zhou and D. De Roure, "Floodnet: Coupling adaptive sampling with energy aware routing in a flood warning system," *Journal of Computer Science and Technology*, vol. 22, pp. 121-130, 2007.

[5] C. Alippi, *et al.*, "Energy management in wireless sensor networks with energy-hungry sensors," *Instrumentation & Measurement Magazine, IEEE*, vol. 12, pp. 16-23, 2009.

[6] G. Anastasi, *et al.*, "Energy conservation in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 7, pp. 537-568, 2009.

[7] *IU-ATC*. Available: http://www.iu-atc.com/

[8] *Air pollution*. Available: http://www.who.int/topics/air_pollution/en/

[9] S. Vardoulakis, *et al.*, "Modelling air quality in street canyons: a review," *Atmospheric environment*, vol. 37, pp. 155-182, 2003.

[10] R. Berkowicz, *et al.*, "Modelling traffic pollution in streets," *National Environmental Research Institute, Roskilde, Denmark*, vol. 10129, p. 20, 1997.

[11] *Operational street pollution model*. Available: http://www.dmu.dk/en/air/models/ospm/

[12] S. Karra, *et al.*, "The dispersion of traffic related pollutants across a non-homogeneous street canyon," *Procedia Environmental Sciences,* vol. 4, pp. 25-34, 2011.

[13] S. Karra and L. Malki-Epshtein, "Influence of local parameters on the dispersion of traffic-related pollutants within street canyons," in *APS Division of Fluid Dynamics Meeting Abstracts*, 2011.

[14] *Air quality planning and standards*. Available: http://www.epa.gov/air/oaqps/montring.html

[15] B. Croxford, *et al.*, "Real time carbon monoxide measurements from 270 UK homes," 2006.

[16] *Mobile Environmental Sensing System Across Grid Environments*. Available: http://bioinf.ncl.ac.uk/message/

[17] Y. Ma, *et al.*, "Air Pollution Monitoring and Mining Based on Sensor Grid in London," *Sensors,* vol. 8, pp. 3601-3623, 2008.

[18] R. N. Murty, *et al.*, "Citysense: An urban-scale wireless sensor network and testbed," in *Technologies for Homeland Security, 2008 IEEE Conference on*, 2008, pp. 583-588.

[19] L. Cordova-Lopez, *et al.*, "Online vehicle and atmospheric pollution monitoring using GIS and wireless sensor networks," in *Journal of Physics: Conference Series*, 2007, p. 012019.

[20] Y. J. Jung, *et al.*, "Air pollution monitoring system based on geosensor network," in *Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International*, 2008, pp. III-1370-III-1373.

[21] K. K. Khedo, *et al.*, "A wireless sensor network air pollution monitoring system," *arXiv preprint arXiv:1005.1737,* 2010.

[22] K.-J. Wong, *et al.*, "Environmental monitoring using wireless vehicular sensor networks," in *Wireless Communications, Networking and Mobile Computing, 2009. WiCom'09. 5th International Conference on*, 2009, pp. 1-4.

[23] F. Gil-Castineira, *et al.*, "Urban pollution monitoring through opportunistic mobile sensor networks based on public transport," in *Computational*

Intelligence for Measurement Systems and Applications, 2008. CIMSA 2008. 2008 IEEE International Conference on, 2008, pp. 70-74.

[24] L. Cheng, *et al.*, "A wearable and flexible Bracelet computer for on-body sensing," in *Consumer Communications and Networking Conference (CCNC), 2011 IEEE*, 2011, pp. 860-864.

[25] L. V. Shum, *et al.*, "On the Development of a Sensor Module for Real-Time Pollution Monitoring," in *Information Science and Applications (ICISA), 2011 International Conference on*, 2011, pp. 1-9.

[26] L. V. Shum, *et al.*, "Making sense of sensor data in practical wireless sensor network designs," in *Computing, Networking and Communications (ICNC), 2012 International Conference on*, 2012, pp. 187-191.

[27] A. Dunkels, *et al.*, "Contiki-a lightweight and flexible operating system for tiny networked sensors," in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, 2004, pp. 455-462.

[28] A. Dunkels, "Rime-a lightweight layered communication stack for sensor networks," 2007.

[29] A. H. Marcus, "Air pollutant averaging times: notes on a statistical model," *Atmospheric Environment (1967)*, vol. 7, pp. 265-270, 1973.

[30] R. I. Larsen, "A new mathematical model of air pollutant concentration averaging time and frequency," *Journal of the Air Pollution Control Association*, vol. 19, pp. 24-30, 1969.

[31] *The R project for Statistical Computing*. Available: http://www.r-project.org/

[32] C. Chatfield, *The analysis of time series: an introduction* vol. 59: Chapman and Hall/CRC, 2003.

[33] J. Beran, *Statistics for long-memory processes* vol. 61: CRC Press, 1994.

[34] C. K. Peng, *et al.*, "Quantification of scaling exponents and crossover phenomena in nonstationary heartbeat time series," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 5, pp. 82-87, 1995.

[35] J. W. Kantelhardt, *et al.*, "Multifractal detrended fluctuation analysis of nonstationary time series," *Physica A: Statistical Mechanics and its Applications,* vol. 316, pp. 87-114, 12/15/ 2002.

[36] E. A. Ihlen, "Introduction to multifractal detrended fluctuation analysis in matlab," *Front Physiol,* vol. 3, p. 141, 2012.

[37] C.-K. Lee, "Multifractal Characteristics in Air Pollutant Concentration Time Series," *Water, Air, and Soil Pollution,* vol. 135, pp. 389-409, 2002/03/01 2002.

[38] M. Bauer, *et al.*, "Cause and effect analysis of chemical processes analysis of a plant-wide disturbance," in *Control Loop Assessment and Diagnosis, 2005. The IEE Seminar on (Ref. No. 2005/11008)*, 2005, pp. 23-30.

[39] M. Bauer, *et al.*, "Measuring cause and effect between process variables," in *Proceedings of the the IEEE Advanced Process Control Applications for Industry Workshop*, 2005.

[40] M. Bauer, *et al.*, "Nearest Neighbors Methods for Root Cause Analysis of Plantwide Disturbances," *Industrial & Engineering Chemistry Research,* vol. 46, pp. 5977-5984, 2007/08/01 2007.

[41] U. Feldmann and J. Bhattacharya, "Predictability improvement as an asymmetrical measure of interdependence in bivariate time series," *International Journal of Bifurcation and Chaos,* vol. 14, pp. 505-514, 2004.

[42] U. Feldmann and J. Bhattacharya, "Mixed predictability as an asymmetrical measure of interdependence in multivariate time series," in *6th Experimental Chaos Conf., Potsdam, Germany*, 2001.

[43] C.-K. Lee and S.-C. Lin, "Chaos in air pollutant concentration (APC) time series," *Aerosol and Air Quality Research,* vol. 8, pp. 381-391, 2008.

[44] A. B. Chelani, *et al.*, "Nonlinear dynamical characterization and prediction of ambient nitrogen dioxide concentration," *Water, air, and soil pollution,* vol. 166, pp. 121-138, 2005.

[45] J.-L. Chen, *et al.*, "Nonlinear dynamics of hourly ozone concentrations: nonparametric short term prediction," *Atmospheric environment,* vol. 32, pp. 1839-1848, 1998.

[46] R. Foxall, *et al.*, "On nonlinear processing of air pollution data," in *Artificial Neural Nets and Genetic Algorithms*, 2001, pp. 477-480.

[47] M. Lanfredi and M. Macchiato, "Searching for low dimensionality in air pollution time series," *EPL (Europhysics Letters)*, vol. 40, p. 589, 1997.

[48] H. Kantz and T. Schreiber, *Nonlinear time series analysis*, 2nd ed. Cambridge, UK ; New York: Cambridge University Press, 2004.

[49] A. Schmitz, "Measuring statistical dependence and coupling of subsystems," *Physical review. E, Statistical physics, plasmas, fluids, and related interdisciplinary topics*, vol. 62, pp. 7508-7511, 2000.

[50] R. Q. Quiroga, *et al.*, "Learning driver-response relationships from synchronization patterns," *Physical Review E*, vol. 61, p. 5142, 2000.

[51] J. Theiler, *et al.*, "Testing for nonlinearity in time series: the method of surrogate data," *Physica D: Nonlinear Phenomena*, vol. 58, pp. 77-94, 1992.

[52] *Complementary error function*. Available: http://www.mathworks.co.uk/help/matlab/ref/erfc.html

[53] C. Alippi, *et al.*, "Adaptive sampling for energy conservation in wireless sensor networks for snow monitoring applications," in *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on*, 2007, pp. 1-6.

[54] C. Alippi and M. Roveri, "An adaptive CUSUM-based test for signal change detection," in *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, 2006, p. 4 pp.

[55] A. Jain and E. Y. Chang, "Adaptive sampling for sensor networks," in *Proceeedings of the 1st international workshop on Data management for sensor networks: in conjunction with VLDB 2004*, 2004, pp. 10-16.

[56] J. J. LaViola Jr, "An experiment comparing double exponential smoothing and Kalman filter-based predictive tracking algorithms," in *Virtual Reality, 2003. Proceedings. IEEE*, 2003, pp. 283-284.

[57] H. Liu, *et al.*, "eSENSE: energy efficient stochastic sensing framework scheme for wireless sensor platforms," in *Proceedings of the 5th international conference on Information processing in sensor networks*, 2006, pp. 235-242.

[58] H. Liu, *et al.*, "dsense: Data-driven stochastic energy management for wireless sensor platforms," *Dept. of CSE, Univ. of Minnesota, Tech. Rep. TR*, pp. 05-018, 2005.

[59] A. Deshpande, *et al.*, "Model-driven data acquisition in sensor networks," in *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, 2004, pp. 588-599.

[60] P. Padhy, *et al.*, "A utility-based adaptive sensing and multihop communication protocol for wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, p. 27, 2010.

[61] D. Tulone and S. Madden, "PAQ: Time series forecasting for approximate query answering in sensor networks," in *Wireless Sensor Networks*, ed: Springer, 2006, pp. 21-37.

[62] D. Tulone and S. Madden, "An energy-efficient querying framework in sensor networks for detecting node similarities," in *Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems*, 2006, pp. 191-300.

[63] C. Liu, *et al.*, "Energy efficient information collection with the ARIMA model in wireless sensor networks," in *Global Telecommunications Conference, 2005. GLOBECOM'05. IEEE*, 2005, pp. 5 pp.-2474.

[64] D. J. Wright, "Forecasting data published at irregular time intervals using an extension of Holt's method," *Management Science*, vol. 32, pp. 499-510, 1986.

[65] Q. Ye, *et al.*, "Short-term traffic speed forecasting based on data recorded at irregular intervals," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, 2010, pp. 1541-1546.

[66] G. Werner-Allen, *et al.*, "Monitoring volcanic eruptions with a wireless sensor network," in *Wireless Sensor Networks, 2005. Proceeedings of the Second European Workshop on*, 2005, pp. 108-120.

[67] *TriScatteredInterp*. Available: http://www.mathworks.co.uk/help/matlab/ref/triscatteredinterp.html

[68] S. Fortune, "Voronoi diagrams and Delaunay triangulations," *Computing in Euclidean geometry*, vol. 1, pp. 193-233, 1992.

[69] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer communications*, vol. 30, pp. 2826-2841, 2007.

[70] W. R. Heinzelman, *et al.*, "Energy-efficient communication protocol for wireless microsensor networks," in *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, 2000, p. 10 pp. vol. 2.

[71] A. Meka and A. K. Singh, "Distributed spatial clustering in sensor networks," in *Advances in Database Technology-EDBT 2006*, ed: Springer, 2006, pp. 980-1000.

[72] J. Yin and M. M. Gaber, "Clustering distributed time series in sensor networks," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, 2008, pp. 678-687.

[73] T. Li, *et al.*, "A survey on wavelet applications in data mining," *ACM SIGKDD Explorations Newsletter*, vol. 4, pp. 49-68, 2002.

[74] N. Vlajic and D. Xia, "Wireless sensor networks: to cluster or not to cluster?," in *Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks*, 2006, pp. 258-268.

[75] X. Ma, *et al.*, "Distributed, hierarchical clustering and summarization in sensor networks," in *Advances in Data and Web Management*, ed: Springer, 2007, pp. 168-175.

[76] X.-L. Ma, *et al.*, "DHC: Distributed, Hierarchical Clustering in Sensor Networks," *Journal of Computer Science and Technology*, vol. 26, pp. 643-662, 2011.

[77] C.-H. Lung and C. Zhou, "Using hierarchical agglomerative clustering in wireless sensor networks: An energy-efficient and flexible approach," *Ad Hoc Networks*, vol. 8, pp. 328-344, 2010.

[78] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *science*, vol. 315, pp. 972-976, 2007.

[79] *Affinity propagation*. Available: http://www.psi.toronto.edu/index.php?q=affinity%20propagation

[80] S. Datta, *et al.*, "K-Means Clustering Over a Large, Dynamic Network," in *SDM*, 2006.

[81] P. Szczytowski, *et al.*, "Asample: Adaptive spatial sampling in wireless sensor networks," in *Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC), 2010 IEEE International Conference on*, 2010, pp. 35-42.

[82] R. Willett, *et al.*, "Backcasting: adaptive sampling for sensor networks," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*, 2004, pp. 124-133.

[83] S. Lin, *et al.*, "Region sampling: Continuous adaptive sampling on sensor networks," in *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, 2008, pp. 794-803.

[84] E. K. Lee, *et al.*, "SILENCE: distributed adaptive sampling for sensor-based autonomic systems," in *Proceedings of the 8th ACM international conference on Autonomic computing*, 2011, pp. 61-70.

[85] B. Gedik, *et al.*, "ASAP: an adaptive sampling approach to data collection in sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 18, pp. 1766-1783, 2007.

[86] *One sample and paired sample t-test*. Available: http://www.mathworks.co.uk/help/stats/ttest.html

[87] C. Liu, *et al.*, "An energy-efficient data collection framework for wireless sensor networks by exploiting spatiotemporal correlation," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 18, pp. 1010-1023, 2007.

[88] H. Adorf, "Interpolation of irregularly sampled data series–a survey," *Astronomical Data Analysis Software and Systems IV*, vol. 77, p. 460, 1995.

[89] P. M. Broersen, "Time series analysis for irregularly sampled data," in *Proc. IFAC World Conf*, 2005.

[90] J. N. Pan and S. T. Chen, "Monitoring long-memory air quality data using ARFIMA model," *Environmetrics*, vol. 19, pp. 209-219, 2008.

[91] L. Y. Siew, *et al.*, "ARIMA and Integrated ARFIMA models for forecasting air pollution index in shah alam, selangor," *The Malaysian Journal of Analytics Sciences,* vol. 12, pp. 257-263, 2008.

[92] T. Schreiber, "Measuring Information Transfer," *Physical Review Letters,* vol. 85, pp. 461-464, 07/10/ 2000.

[93] M. Bauer, *et al.*, "Finding the Direction of Disturbance Propagation in a Chemical Process Using Transfer Entropy," *Control Systems Technology, IEEE Transactions on,* vol. 15, pp. 12-21, 2007.

[94] H. Bayraktar and F. S. Turalioglu, "A Kriging-based approach for locating a sampling site—in the assessment of air quality," *Stochastic Environmental Research and Risk Assessment,* vol. 19, pp. 301-305, 2005.